

Maria Nyqvist

IT-yrityksen projektitoiminta

Metropolia Ammattikorkeakoulu
Insinööri (AMK)
Mediatekniikan koulutusohjelma
Insinöörityö
2.5.2011

Tekijä Otsikko	Maria Nyqvist IT-yrityksen projektitoiminta
Sivumäärä Aika	40 sivua + 1 liite 2.5.2011
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen tekniikka
Ohjaajat	palvelupäällikkö Pia Pollari lehtori Merja Bauters
<p>Insinööriyön tarkoituksena oli tutkia tapauksenhallintajärjestelmän käyttöönottoa ja sen kehitysprojektin etenemistä IT-yrityksessä. Työn tavoitteena oli saada vaatimusmäärittely tehtyä, sen pohjalta toteuttaa tarvittavat vaatimukset järjestelmään ja samalla seurata projektin etenemistä.</p> <p>Projekti käynnistettiin syyskuussa 2010. Projektin alkuvaiheessa asetettiin järjestelmän käyttöönottoaikatauluksi marraskuu 2010. Ensimmäinen vaihe oli vaatimusmäärittelyn teko. Siinä apuna käytettiin asiakastytyväisyyskyselyn tuloksia ja verrattiin vanhaa ja uutta järjestelmää keskenään. Järjestelmään oli tarkoitus tehdä asiakaskäyttöliittymä ja työntekijöiden käyttöliittymä, joiden vaatimukset määriteltiin erikseen.</p> <p>Vaatimusmäärittelyn jälkeen oli tarkoitus aloittaa vaatimusten toteutus ja niiden testaaminen. Työssä tutkittavan järjestelmän kehitysprojektin seurantaan osallistui kehitystiimi, johon kuului insinööriyön tekijän lisäksi projektipäällikkö ja tekninen asiantuntija, joka oli samalla mukana myös asiakasprojekteissa. Lokakuussa 2010 järjestelmän käyttöönottoa jouduttiin kuitenkin lykkäämään uusien asiakasprojektien myötä.</p> <p>Nykyään projektitoiminta on yleistynyt yrityksissä. Asiakasprojektit ovat yrityksille tulon lähde, kun taas sisäiset projektit ovat täytettä asiakasprojektien välillä. Projektin aikana oli selvästi havaittavissa, miten asiakasprojekti menee lähes aina sisäisen projektin edelle. Työssä tutkittiin myös projektitoiminnan ketteriä toimintatapoja, koska nykyajan projektitoiminta pitäisi saada niin sisäisesti kuin ulkoisesti toimimaan ketterästi. Menetelmät ovat yleistyneet viime vuosien aikana, ja näitä toimintatapoja voi harvoin hyödyntää yrityksen sisäisiin projekteihin, mutta asiakasprojekteissa ne ovat hyvin vartenotettavia.</p> <p>Insinööriyön palautushetkellä projekti on edelleen samassa pisteessä kuin lokakuussa 2010. Sen verran projekti on edennyt, että kehittäjä on tehnyt vaatimusmäärittelyn pohjalta teknisen suunnitelman, miten vaadittavat ominaisuudet voitaisiin toteuttaa järjestelmään. Työssä tutkittavan projektin tulevaisuus on työntäyteinen. Suurimmat askeleet projektin aikana ovat vielä muutosten toteutus, järjestelmän testaus, vanhojen tietojen tuominen vanhasta järjestelmästä uuteen järjestelmään ja varsinainen käyttöönotto.</p>	
Avainsanat	projektitoiminta, vaatimusmäärittely, ketterät toimintatavat

Author Title	Maria Nyqvist Project management in IT company
Number of Pages Date	40 pages + 1 appendice 2 May 2011
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructors	Pia Pollari, Service Manager Merja Bauters, Senior Lecturer
<p>The aim of this thesis was to investigate the implementation of the incident management system and the progression of the development project in IT company. Aim of the thesis was to get the requirement specification done and execute the requirements. The aim was also to follow the progression of the development project at the same time.</p> <p>The project was started in September 2010. The implementation of the system was set to November 2010. The first stage of the project was to set the requirements. They were based on the results of customer satisfaction enquiry and a comparison between the old and new system. There were supposed to be two interfaces: one to the customers and one to employees. The requirements of the interfaces were done separately.</p> <p>After the requirement specification was done the purpose was to start the implementation of the requirements and test them. The development team which participate the development project contains a project manager and a technical developer who was doing some customer projects at the same time. In October 2010 the implementation had to be postponed because of the new customer projects.</p> <p>Nowadays the projects have become a common concept between companies. The customer projects are the source of incomes to companies, whereas the internal projects are a way to fill in time between the customer projects. During the development project it was evident that a customer project almost always goes ahead. The thesis investigates some agile methods because project management should be able to function agile nowadays. The methods have become more common in last few years and these methods cannot often be used in internal projects but in customer projects they work well.</p> <p>When the thesis was returned, the development project was at the same state than in October 2010. The only thing that has been done is technical specification. It was made based on the requirement specification. The future of the projects is includes lot of work. The biggest steps are changes, testing of the system, importing the old data from the old system to the new system and the actual implementation.</p>	
Keywords	Project management, requirements specification, agile methods,

Sisällys

1	Johdanto	1
2	Projektitoiminta ja projektin aloitus	2
3	Tietoteknisen järjestelmän vaatimusmäärittely	4
4	Tietoteknisen järjestelmän suunnittelu	12
4.1	Toiminnallisuus	12
4.2	Rakenne	15
4.3	Järjestelmän käyttöliittymä	17
4.3.1	Käyttöliittymän elementit	17
4.3.2	Visuaalinen suunnittelu	19
5	Tapauksenhallintajärjestelmästä puuttuvat ominaisuudet	22
5.1	Vasteajat	22
5.2	Jononäkymä	25
5.3	Raportointi	26
6	Projektin kulku	27
6.1	Projektin eteneminen ja ketterät toimintatavat	27
6.2	Projektin tulevaisuus	31
6.2.1	Testaus	31
6.2.2	Käyttöönotto	34
7	Sisäisten projektien kehitysehdotus ja johtopäätökset	35
8	Yhteenveto	36
	Lähteet	39
	Liitteet	
	Liite 1. Asiakastyytyväisyyskyselyn tulokset	
	Liite 2. Vasteaikaesimerkin laskukaava	

1 Johdanto

Projektitoiminta voidaan määritellä monella eri tavalla. Projekti on yleisesti hanke tai toimitus, joka on suunniteltu huolellisesti ja jonka lopputuloksena on konkreettinen tuote tai palvelu. Projekti voi olla yrityksessä sisäinen tai ulkoinen eli asiakasprojekti. Projektitoiminta on yleistynyt yrityksissä, ja siihen halutaan panostaa. Myös projektitoimintaan liittyvät ketterät menetelmät ovat yleistyneet. Niitä esittelen myöhemmin työssä.

Insinööriyön tarkoituksena on tutkia IT-yrityksen tapauksenhallintajärjestelmän käyttöliittymän suunnittelua ja toteutusta sekä puuttuvien osien upotusta olemassa olevaan järjestelmään. Työssä pohditaan myös vaatimusmäärittelyn tärkeyttä, järjestelmän tarkkaa suunnittelua ja valmiin järjestelmän testausta sekä kehitysprojektin kulkua ja sen onnistumista ketterien menetelmien avulla. Työssä esitellään järjestelmän vaatimusmäärittely, projektin kulku ja kehitysehdotuksia.

Työn tavoite on määritellä järjestelmän puuttuvat osat ja priorisoida ne, tutkia projektin etenemistä, sen onnistumista ja projektitoimintaa yleisesti. Johtopäätöksissä nojautaan esiteltyyn teoriaan ja omaan kokemukseen. Työn tuotoksena on tarkoituksena saada aikaan järjestelmän määrittelydokumentti ja arviointi projektin onnistumisesta ja mahdollisista kehitysajatuksista.

Työ tehdään tietotekniikan palveluyritykselle. Se tuottaa asiakkaidensa liiketoiminnan menestykselle välttämättömiä järjestelmiä. Yrityksen strategisen kumppanin asema nojaa luottamukseen, innovatiivisuuteen, ketteryyteen, korkeaan laatuun ja kestävään kasvuun. Yrityksen järjestelmät perustuvat tarkoituksenmukaisiin teknologioihin ja kumppanuuksiin, sillä jokainen bisnes on omanlaisensa. Yrityksessä työskentelee noin 170 ammattilaista Espoossa, Kuopiossa, Lappeenrannassa ja Pietarissa.

Työssä tutkittava tapauksenhallintajärjestelmä on tärkeä osa yrityksen ylläpitotoimintaa. Ensisijaisen tärkeä se on asiakaspalvelijoille, palvelupäälliköille ja ylläpidon teknisille asiantuntijoille. Järjestelmään kirjataan ylläpidon asiakkailta tulleita vikailmoituksia, kysymyksiä, ongelmatapauksia ja muutoksia, jotka asiakas voi kirjata itse tai ne voi

kirjata asiakaspalvelija. Lähes kaikki kommunikaatio asiakkaan ja asiakaspalvelun välillä hoidetaan järjestelmän kautta. Asiakaspalvelun tärkein tehtävä on seurata kaikkien ylläpitoasiakkaiden kaikkia avoinna olevia tapauksia ja niiden etenemistä. Se tehdään järjestelmän jononäkymässä, johon on listattu kaikki avoinna olevat tapaukset. Tärkeä asia järjestelmässä on myös vasteaikojen eteneminen, niiden seuraaminen ja mahdollinen ylittyminen. Vasteaika-käsite selitetään tarkemmin luvussa 5.1.

2 Projektitoiminta ja projektin aloitus

Projekti voidaan määritellä monella eri tavalla. Se voi olla lyhyt tai pitkä, helppo tai vaikea, yrityksen sisäinen tai ulkoinen. Yleisin mittapuu projekteissa on kuitenkin raha. Kaikkia projekteja yhdistää se, että niissä on määritelty päämäärä, alku ja loppu. Projektille ominaisia tunnuspiirteitä ovat tavoitteellisuus, suunnitelmallisuus, kertaluonteisuus ja määrätyt resurssit, kuten tekninen asiantuntija ja projektipäällikkö. Projektin tuotos voi olla mitä tahansa, ja tässä työssä on keskitytty vain tietotekniikan ja tarkemmin järjestelmäkehitysprojektiin. (1.)

Projektitoiminta on yleistynyt IT-yrityksissä. Syitä yleistymiseen on monia, muun muassa yhteiskunnallinen muutos, työn ja resurssien tehokkuuden maksimointi ja siirtyminen perusrahoituksesta hankerahoitukseen. Yhteiskunnallisella muutoksella tarkoitetaan verkostoitumista ja riippuvuutta asiantuntijoista, mikä on lisääntynyt viime vuosien aikana. Työn ja resurssien tehokkuuden maksimointi takaa sen, että projekti tehdään aikataulussa ja budjetissa pysyen ja hankerahoituksella taataan rahoitus koko hankkeelle. (1.)

Projekti voidaan jakaa neljään eri vaiheeseen: suunnitteluun, käynnistämiseen, toteutukseen ja päättämiseen. Suunnitteluvaihe on projektin tärkein vaihe: hyvin suunniteltu on jo puoliksi tehty. Kuitenkin ennen varsinaista projektin suunnitteluvaihetta on syytä tarkistaa resurssien tilanne. Resurssit koostuvat tekijöistä, ajasta sekä taloudellisista ja materiaalisista resursseista. Nämä on oltava suunniteltu ja kunnossa, ennen kuin projekti aloitetaan. (1.)

Kun resurssit on tarkistettu, seuraava askel on projektisuunnitelman ja -määrittelyn teko. Suunnitteluvaiheen tärkein tavoite on tarkentaa projektin tehtävänantoa ja tavoit-

teita sekä aikatauluttaa projektin eri vaiheet. Suunnitteluvaiheessa on myös hyvä käydä läpi mahdollisia riskejä, joita liittyy projektiin ja sen etenemiseen. Jos projekti voidaan tuottaa monella eri tavalla, on suunnitteluvaiheessa syytä valita myös toteuttamistekniikka. Yksi ja ehkä olennaisin asia projektin suunnittelussa usein unohtuu: projektisuunnitelma päivitys projektin edetessä. Projektisuunnitelmassa tulisi olla ainakin seuraavat asiat:

- projektin tavoitteet ja niiden priorisointi
- projektiorganisaatio eli projektiin osallistuvat henkilöt ja ryhmät sekä niiden valtuudet, vastuut ja raportointilinjat
- projektin valvonta ja ohjaus sekä niihin liittyen projektisuunnitelman ylläpito
- projektin dokumentointi sekä tuotoksen että projektin läpiviennin osalta
- sisäinen ja ulkoinen raportointi
- projektissa suoritettavat tehtävät ja niiden riippuvuudet ja vaiheistus
- projektin aikataulu
- projektin henkilöresurssien käyttö ja työnjako
- projektin muut resurssit (esimerkiksi tilat ja laitteet)
- projektissa käytetyt menetelmät ja työkalut
- riskienhallinta
- laadunvarmistus. (2 s. 56–57.)

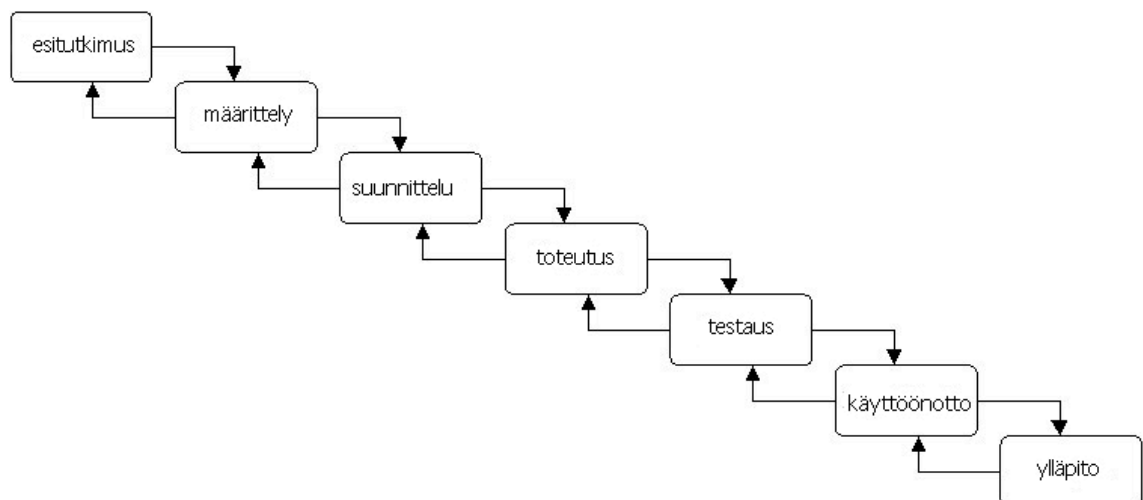
Insinööriyössä tutkittu projekti aloitettiin syyskuussa 2010. Projektin alkuperäinen ajatus oli yhtenäistää yrityksen käytössä olevat tapauksenhallintajärjestelmät, koska yrityksessä oli aikaisemmin elokuussa ollut suuria yritysjärjestelyitä, mikä oli syynä kahden eri järjestelmän käyttöön. Projektin tavoitteena oli kaikkien tarpeellisten ominaisuuksien huomioon ottaminen ja niiden saaminen valittuun järjestelmään. Heti projektin alkuvaiheessa muutama tärkeä puuttuva ominaisuus nousi yli muiden, joten priorisointi tuli kuin itsestään.

Projektiin osallistuvat henkilöt päätettiin heti projektin alussa. Se ei tuottanut ongelmaa, koska yrityksessä on vain muutama valitun tekniikan osaaaja. Projektin aikataulun sijaan tuotti enemmänkin haasteita. Alkuperäinen järjestelmän käyttöönotto sovittiin projektin alkaessa marraskuun alkuun 2010. Se todettiin kuitenkin aivan liian hätäiseksi arvioksi. Tarve uudelle järjestelmälle oli kova ja tahtotila löytyi, mutta kun kyseessä oli yrityksen sisäinen projekti, oli sille vaikeaa löytää aikaa. Asiakasprojektit me-

nevät lähes aina sisäisten projektin edelle, koska niistä yritys saa tuloja. Tilannetta voisi auttaa, jos sisäisen projektin kulut laskettaisiin ja verrattaisiin saataviin tuloihin. Jos kuitenkin järjestelmä halutaan käyttöön, jossakin vaiheessa sen kehitys on priorisoitava asiakasprojekteja korkeammalle. Pitkällä tähtäimellä yritys saattaa jopa kärsiä taloudellisesti järjestelmien moninaisuudesta, koska työntekijät joutuvat poukkoilemaan järjestelmien välillä, mikä hankaloittaa päivittäistä työtä.

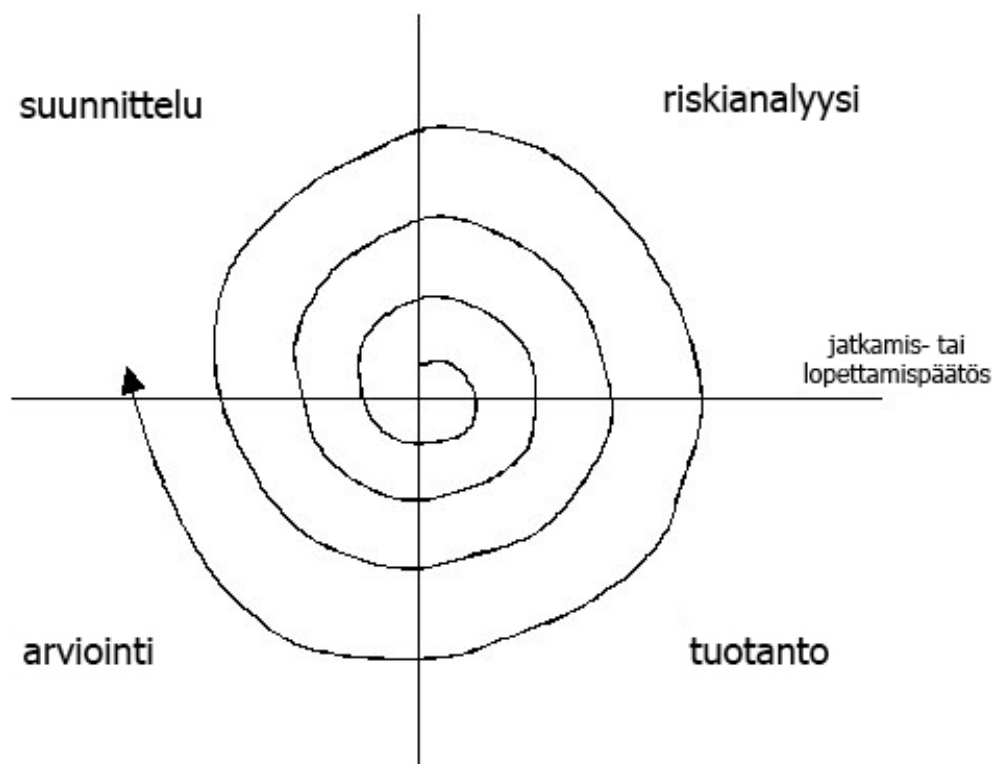
3 Tietoteknisen järjestelmän vaatimusmäärittely

Järjestelmän elinkaari koostuu useista eri vaiheista. Elinkaarimalleja on erilaisia, muun muassa vesiputousmalli, spiraalimalli ja prototyyppilähestymistapa. On hyvä kuitenkin muistaa, että kyseessä on vain malli, jonka sopimista kohde organisaatioon tai sovel-lusalueeseen täytyy järjestelmäkohtaisesti miettiä. Kuvassa 1 on nähtävissä vesipu-tousmalli, joka oli ensimmäinen elinkaarimalli. Se kehitettiin 1960-luvulla perinteisten fyysisten prosessimallien pohjalta. Vesiputousmallissa järjestelmän eri vaiheet seuraa-vat suoraviivaisesti toisiaan, mikä vaikeuttaa taaksepäin palaamista. Lähes jokaisessa projektissa tulee esille edellisessä vaiheessa tehtyjä virheitä, mikä vaatii taaksepäin palaamista ja virheiden korjaamista. Tämän prosessin vesiputousmalli kuvaa huonosti. (2, s. 40.)



Kuva 1: Tietoteknisen järjestelmän elinkaaren vesiputousmalli (3).

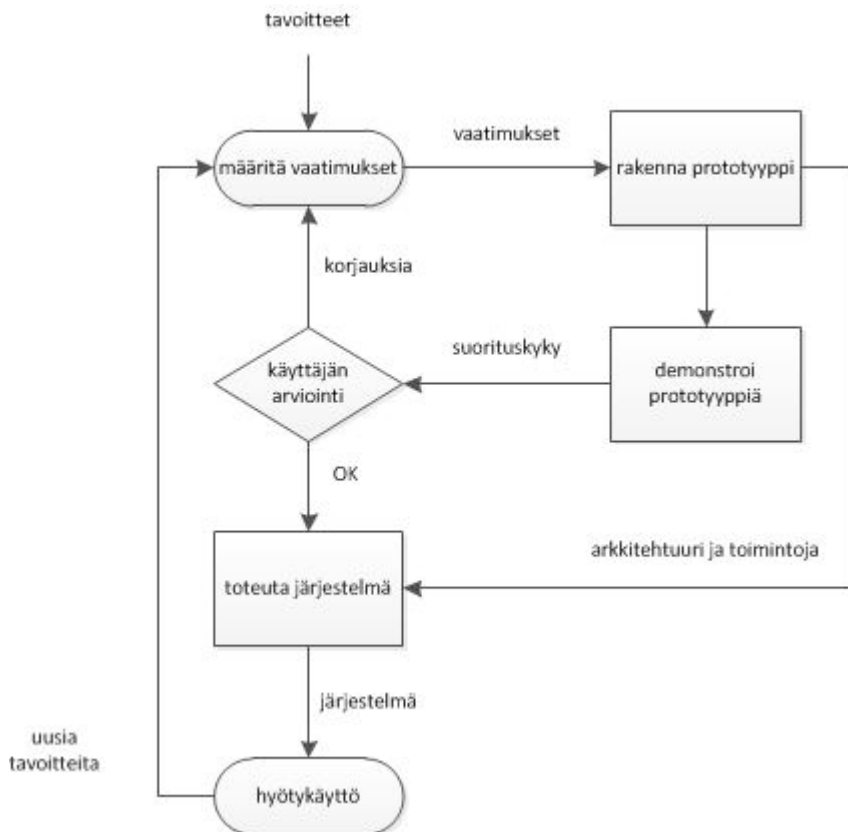
Spiraalimalli on toinen elinkaarimalli, ja se eroaa vesiputousmallista iteratiivisuuden osalta. Iteratiivisuudella tarkoitetaan mahdollisuutta peruttaa edellisiin vaiheisiin korjaamaan mahdolliset virheet. Iteratiivisuuden lisäksi spiraalimallissa toinen keskeinen ajatus on riskien analysointi. Muissa elinkaarimalleissa tätä ei juuri ole otettu huomioon, mikä tekee spiraalimallista erilaisen. Kuvassa 2 on nähtävissä spiraalimallin neljä eri vaihetta: suunnittelu, riskianalyysi, tuotanto ja asiakkaan suorittama arviointi. Näitä neljää vaihetta toistetaan koko järjestelmän elinkaaren ajan. Spiraalimalli on uusi elinkaarimalli, eikä sen käytöstä näin ollen ole yhtä paljon kokemuksia kuin vesiputousmallin tai prototyyppilähestymistavan käytöstä. (2, s. 42–43.)



Kuva 2: Tietoteknisen järjestelmän elinkaari spiraalimalli (3).

Kolmas esiteltävä elinkaarimalli on prototyyppilähestymistapa. Tämä malli sopii projekteihin, joissa asiakas haluaa nopeasti päästä testaamaan järjestelmää tai saada kuvan järjestelmän toimivuudesta. Järjestelmän prototyyppi on kevyt versio, jossa on vain järjestelmän yleiset toiminnallisuudet, mutta ei mitään yksityiskohtia. Prototyyppilähestymistavan periaate on nähtävissä kuvassa 3. Ensimmäisenä asetetaan asiakkaan vaatimukset järjestelmälle, minkä pohjalta rakennetaan ensimmäinen prototyyppi. Prototyypin valmistuttua asiakas testaa sitä ja arvioi sen käytettävyyden. Asiakkaan arvioinnin pohjalta prototyyppiä parannetaan ja arvioidaan uudestaan niin kauan, kunnes

asiakas on tyytyväinen. Lopuksi järjestelmä toteutetaan prototyypin pohjalta. (2, s. 41.)



Kuva 3: Tietoteknisen järjestelmän elinkaaren protoryypilähestymistapa (2, s. 41).

Työssä tutkittavassa järjestelmän käyttöönotossa käytettiin prototyypilähestymistapaa. Projektin aikana tehtiin ensin määrittely, jonka pohjalta on tarkoitus tehdä järjestelmän prototyyppi. Järjestelmästä puuttui muutama tärkeä ominaisuus, joten on tärkeää päästä testaamaan niiden toimivuutta ennen käyttöönottoa.

Vaatusmäärittely on tärkeässä roolissa järjestelmän kehityksessä ja sen elinkaaren aikana. Kun uusi järjestelmä päätetään ottaa käyttöön, on hyvä alusta lähtien selvittää, mitä siihen halutaan, ja tätä varten on vaatusmäärittely. Vaatusmäärittelyllä tarkoitetaan dokumenttia, johon kootaan ominaisuuksia, joita järjestelmään halutaan ja tarvitaan. Dokumenttiin on syytä kirjoittaa myös vaatimusten prioriteetit, koska se helpottaa niiden toteuttamisjärjestystä. Vaatusmäärittelyvaiheen tarkoitus on saada mielikuvat paperille ja siitä eteenpäin toteutettavaksi. On kuitenkin hyvä muistaa, että vaikka asiakkaan mielikuvat saataisiinkin kuvattua vaatusmäärittelyyn tarkasti, se ei tuo hänelle varmuutta tulevan järjestelmän ulkonäöstä ja toimivuudesta. Siksi onkin

hyvä käydä vaatimusmäärittely asiakkaan kanssa läpi niin, että asiakas ymmärtää, mitä dokumentissa olevilla asioilla tarkoitetaan, ja lopuksi hyväksyttää dokumentti asiakkaalla allekirjoituksin. (4.)

Verkkopalvelut ja -järjestelmät ovat haastavia määrittelyn suhteen. Mukaan haluttaisiin eri toiminnallisuuden, ja kaiken pitäisi olla hyvännäköistäkin, mutta vaatimukset eivät aina kohtaa todellisuutta. Vaatimusmäärittelyn verkkopalveluiden ja -järjestelmien osalta pitäisi vastata seuraaviin kysymyksiin:

- Toiminnalliset vaatimukset: Mitä palvelu tekee?
- Tietovaatimukset: Mitä tietoja järjestelmään tarvitaan?
- Toimintaympäristövaatimukset: Minkälaisia laitteita käytetään järjestelmän käyttämiseen?
- Käyttäjävaatimukset: Minkälaisia ominaispiirteitä, tietotasoa ja kokemuksia järjestelmän peruskäyttäjällä on?
- Käytettävyyksivaatimukset: Onko palvelu tehokas, turvallinen, helppokäyttöinen?
- Saavutettavuusvaatimukset: Miten on huomioitu eri ikä- ja/tai vammaisryhmät?
- Turvallisuusvaatimukset: Miten tietoturva on huomioitu järjestelmässä?
- Projektiin liittyvät vaatimukset: Minkälaiset ovat projektin aikataulu ja kustannukset? (5, s. 49.)

Vaatimukset voidaan jakaa karkeasti kahteen eri ryhmään: toiminnallisiin ja ei-toiminnallisiin. Toiminnalliset vaatimukset kertovat, mitä järjestelmän pitäisi tehdä. Ne määrittävät, kuinka järjestelmä toimii käyttäjän näkökulmasta tarkasteltuna, miten kommunikointi ympäristön kanssa onnistuu ja miten yhteydet järjestelmän ja sidosryhmien välillä toimivat. Esimerkkinä toiminnallisesta vaatimuksesta on käyttäjän pääsy tarkistamaan omat tietonsa ja tapauksensa. Ei-toiminnalliset vaatimukset puolestaan kertovat reunaehdot, joiden vallitessa järjestelmä täyttää toiminnalliset vaatimukset. Ei-toiminnalliset vaatimukset määrittelevät esimerkiksi käyttäjämääriä, verkkoasetukset, infrastruktuurin ja tietoturvaa. Ei-toiminnallisten vaatimusten erikoistapauksia kutsutaan rajoitteiksi, joiden avulla voidaan määrittää rajoituksia järjestelmälle asetetuille toiminnallisille vaatimuksille. (2, s. 28; 4.)

Määrittelyvaihe on elinkaarimalleissa heti alussa esitutkimuksen jälkeen, ja kaikki sen jälkeen tuleva pohjautuu määrittelyyn. Jos määrittelydokumentti jää puutteelliseksi, järjestelmään saattaa tulla virheitä, joiden korjaus on aikaa vievää ja kallista. Kun määrittely tehdään ensimmäisellä kerralla oikein ja kunnolla, säästetään sekä aikaa että rahaa. Lopullinen dokumentti vaatii paljon työtä, koska vaatimuksia pitää miettiä ja työstää paljon, ennen kuin ne kirjataan dokumenttiin. Vaatimusten pitää olla tarkasti kuvattu ja ymmärrettäviä. (6.) Vaatimusmäärittelyn ja suunnittelun tärkeys tulee esille ketterissä menetelmissä, joita käyn tarkemmin läpi luvussa 6.

Työssä tutkittavan järjestelmän käyttöönotto alkoi määrittelyllä. Määrittely toteutettiin asiakaspalautteiden perusteella sekä vertailemalla käytössä olevaa ja tulevaa järjestelmää. Vaatimukset priorisoitiin ja mietittiin, missä järjestyksessä ne toteutetaan. Jo projektin alussa oli selvää, että uudesta järjestelmästä puuttuu kolme tärkeää ominaisuutta, jotka määriteltiin myös osaksi uutta järjestelmää ja jotka käydään tarkemmin läpi luvussa 5.

Asiakaspalaute oli tärkeässä roolissa määrittelyä tehtäessä. Järjestelmää käyttäville asiakkaille tehdään asiakastyytyväisyyskysely tietyin väliajoin, ja nämä kyselyn vastaukset ovat erittäin tärkeitä ajatellen järjestelmän kehitystä. Asiakastyytyväisyyskyselyn kysymykset ja tulokset löytyvät liitteestä 1. Määrittelyä tehtäessä alusta asti oli selvää, että asiakkaat pitivät selainpohjaista järjestelmää hyödyllisenä eikä siitä haluttu luopua. Myönteistä palautetta vanha järjestelmä sai helppokäyttöisyydestä, visuaalisesta ilmeestä, teknisestä vakaudesta ja käyttöohjeistuksesta. Myös kommunikointi koettiin helpoksi järjestelmän kautta. Ainoa selvästi huono asia käytössä olevassa järjestelmässä on vanhojen tapauksien selailu. Koko järjestelmässä ei ole käytössä minkäänlaista hakua, jolla voisi etsiä vanhoja tapauksia. Myös viestiketjujen huono lukumahdollisuus ja liitetiedostojen tallentamisen vaikeus sai kritiikkiä. Määrittelyssä nämä ominaisuudet todettiin erittäin tarpeellisiksi lisätä uuteen järjestelmään.




Toinen tärkeä osa määrittelyä asiakaspalautteen lisäksi oli uuden ja käytössä olevan järjestelmän vertailu. Uudessa järjestelmässä oli kolme selvää puutetta heti alussa: vasteajat ja raportointi puuttuivat kokonaan ja tapauksien jononäkymä oli puutteellinen. Vasteajat ovat ylläpidon puolella liiketoimintakriittinen ominaisuus, joten niiden saaminen järjestelmään priorisoitiin korkeimmaksi. Myös raportointi on erittäin tärkeä

ominaisuus, koska se perustuu asiakkaan ja toimittajan tekemään tukipalvelusopimukseen, ja näin ollen se oli pakko saada järjestelmään. Jononäkymä uudessa järjestelmässä on erittäin epäkäytännöllinen, mikä vaikeuttaisi asiakaspalvelun jokapäiväistä työtä, ja koska asiakaspalvelu tulee käyttämään järjestelmää eniten, on jononäkymää muutettava sujuvammaksi.

Järjestelmä tulee käyttöön sekä asiakkaille että yrityksen työntekijöille, joten määrittelyssä määriteltiin erikseen asiakaskäyttöliittymän ominaisuudet ja peruskäyttäjien käyttöliittymän ja toiminnallisuuksien ominaisuudet. Asiakaskäyttöliittymässä on tärkeää, että käyttäjä voisi valita itse järjestelmän kielen suomeksi tai englanniksi. Vanhassa järjestelmässä ovat olleet käytössä molemmat kielet samanaikaisesti, mikä on tehnyt järjestelmästä kömpelön. Koska järjestelmä on selainpohjainen, sen on tärkeää tukea useampaa kuin yhtä selainta. Vanha järjestelmä toimi vain Internet Explorerilla, joten uuden järjestelmän tulisi olla selainriippumaton. Myös mahdollisuus tallentaa tunnus ja salasana selaimen muistiin koettiin tärkeäksi asiakkaiden keskuudessa.

Asiakaskäyttöliittymän kautta asiakkaat ilmoittavat vikatilanteista tai kysyvät sovelluksiinsa liittyviä kysymyksiä. On tärkeää, että uutta tapausta luotaessa tulevat esiin kaikki tärkeät tiedot. Tapaukselle valitaan projekti, johon tapaus kuuluu ja johon käyttäjällä on oikeus tehdä tapauksia, prioriteetti (A/B/C/D), tyyppi (vikatilanne/kysymys/pyyntö/muutos), otsikko ja tarkempi kuvaus. Tapaukselle voi myös tallentaa liitteen. Tapauksen voi myös luoda laittamalla sähköpostia suoraan järjestelmän omaan sähköpostiosoitteeseen. Vanha järjestelmä ei tunnistanut sähköpostissa olevia liitteitä, joten uuteen järjestelmään tämä pitää korjata. Kun asiakas avaa vanhan tapauksen, pitäisi hänen nähdä koko tapauksen viestihistoria kerralla ja helposti erottaa omat ja asiakaspalvelun lähettämät viestit (esimerkiksi erilaisten värien avulla). Kuvas-
sa 4 on nähtävissä vanhan järjestelmän ulkoasu ja uuden tapauksen luontilomake.

Uutiset / News
Tapaukset / Incidents
Muutokset / Changes
Profiilini / My Profile
Ohje / Help
Kirjaudu ulos / Log out

Kirjaa tapaus / Log an Incident - New
  

Matti Meikäläinen

Projekt / Project:
Prioriteetti / Priority:
Kuvaus / Description:
Lisätiedot / Details:

OK

Cancel

Kuva 4: Vanhan tapauksenhallintajärjestelmän asiakaskäyttöliittymä.

Kaikki yksittäisen asiakkaan tapaukset ovat nähtävissä listauksessa, jota pitää voida suodattaa ja järjestää projekteittain, prioriteeteittain, statuksittain ja tyypeittäin. Listausnäkylässä pitää näkyä tapauksen ID, kirjauspäivämäärä, projekti, otsikko, status, prioriteetti ja tyyppi. Kuvassa 5 on nähtävissä vanhan järjestelmän asiakaskäyttöliittymän jononäkymä. Tarvittaessa asiakkaan pitäisi saada kaikki haluamansa tapaukset Exceliin. Vanhassa järjestelmässä iso kompastuskivi on ollut hakutoiminnon puuttuminen. Uuteen järjestelmään halutaan haku, jossa tapauksia voi etsiä joko tapauksen ID:llä tai otsikolla.

Tapaukset / Incidents

Tapaukset / Incidents - Test

Status:
Project:
Priority:

#	ID	Avattu / Logged At	Project	Kuvaus / Description	Tila / State	Priority	Tyyppi / Type
1	1005287	25.11.2010 13:41	Testiprojekti	Test ticket for Michael	Resolved	B - Serious	Request
2	1005286	25.11.2010 13:30	Testiprojekti	Test incident for Vitaly	Resolved	B - Serious	Incident
3	1005285	25.11.2010 13:26	Testiprojekti	This is the test incident number 1	Resolved	C - Normal	Incident
4	1004798	27.09.2010 12:40	Testiprojekti	Uusi tiketti	Resolved	C - Normal	Incident
5	1003577	02.05.2010 17:06	Testiprojekti	jees	Transferred	A - Critical	Incident

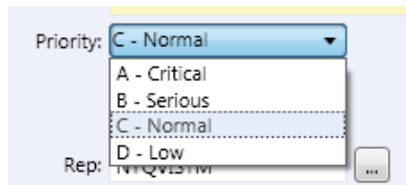
Kuva 5: Vanhan tapauksenhallintajärjestelmän jononäkymä.

Asiakaskäyttöliittymän lisäksi määriteltiin yrityksen työntekijöiden käyttöön tuleva järjestelmä, sen ominaisuudet ja käyttöliittymä. Uutta tapausta tehtäessä valitaan ensin tapauksen tyyppi, status ja asiakaskontakti. Uudesta järjestelmästä puuttuu kokonaan

mahdollisuus valita projektia, mikä on tärkeä ominaisuus, koska useammalla asiakkaalla on monta projektia ja tapaukset pitää kohdistaa oikealle projektille. Kuvassa 6 on vanhan järjestelmän uuden tapauksen luontilomake. Asiakaspalvelun työtä helpottaisi, jos asiakkuuden palvelupääällikkö tulisi aina tapaukselle automaattisesti näkyviin. Uudessa järjestelmässä viestihistoria näkyy paremmin, minkä takia olisi tärkeää myös saada asiakkaan ja asiakaspalvelun lähettämät viestit eroteltua toisistaan, esimerkiksi värein. Silloin näkisi ensimmäisellä vilkaisulla viestihistorian tilanteen.

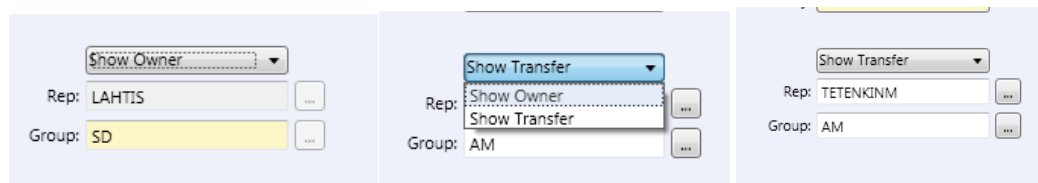
Kuva 6: Vanhan tapauksenhallintajärjestelmän uuden tapauksen luontilomake.

Tapauksen prioriteetti on tällä hetkellä uudessa järjestelmässä väärin. Se pitää korjata asteikolle A–D, joka on asiakassopimuksissa käytetty asteikko. Prioriteetti kertoo tapauksen kiireellisyyden ja töiden tekojärjestyksen. Kuvassa 7 on nähtävissä prioriteetit, jotka pitäisi saada uuteen järjestelmään. Asiakaspalvelun ja teknisen asiantuntijan työtä helpottaisi jokaiseen tapaukseen tuleva asiakastietokenttä. Siihen voisi laittaa asiakkaaseen liittyvät erikoiskäytännöt, mahdolliset sanktiotiedot ja muuten tärkeää tietoa asiakkaaseen liittyen. Uudessa järjestelmässä on mahdollisuus ottaa käyttöön muistutustoiminto. Tapaukselle voi asettaa muistutuksen tietylle päivälle, ja järjestelmä lähettää muistutussähköpostin silloin. Tämä on asiakaspalvelulle tärkeä työkalu, ja se olisi hyvä saada toimimaan.



Kuva 7: Vanhan tapauksenhallintajärjestelmän prioriteettilistaus.

Vanhassa järjestelmässä on ollut käytössä kaksi erilaista tekijää. Ensimmäinen on tapauksen omistaja (owner), joka voi olla joko asiakaspalvelija tai palvelupäällikkö ja joka näkyy asiakkaalle ja kommunikoi asiakkaan kanssa. Toinen on tapauksen tekninen asiantuntija (transfer rep), joka korjaa vikatilanteen tai vastaa asiakkaan kysymykseen ja joka ei ole yhteydessä suoraan asiakkaaseen. Uudessa järjestelmässä on vain yksi tapauksen edistäjä, joka näkyy asiakkaalle. Siihen voisi laittaa asiakaspalvelijan ja teknistä asiantuntijaa varten tehtäisiin uusi tekstikenttä, johon voisi laittaa hänen nimensä. Kuva 8 havainnollistaa, kuinka tapauksen omistajan ja teknisen asiantuntijan valitseminen on ratkaistu vanhassa järjestelmässä.



Kuva 8: Vanhassa tapauksenhallintajärjestelmässä omistajan ja teknisen asiantuntijan valitseminen.

4 Tietoteknisen järjestelmän suunnittelu

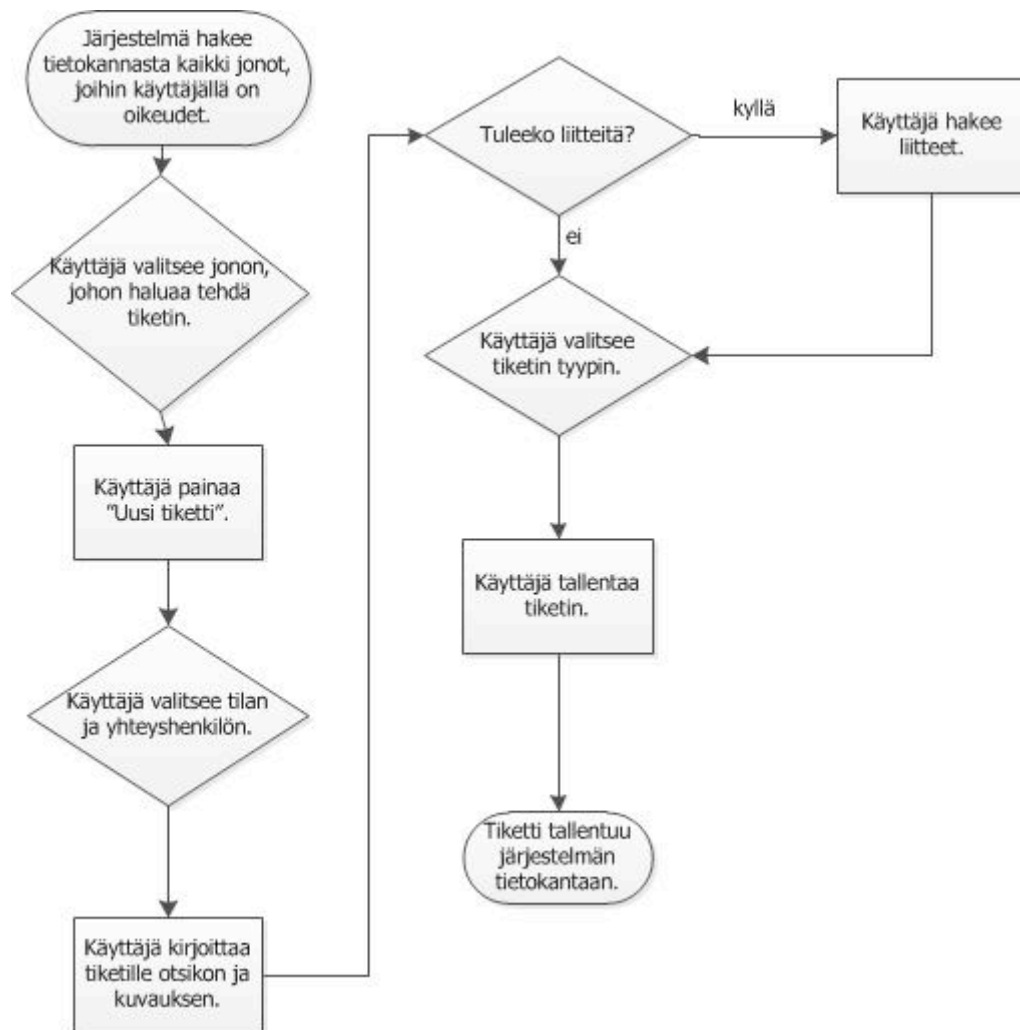
4.1 Toiminnallisuus

Määrittelyn jälkeen seuraava vaihe projektissa on järjestelmän suunnittelu. Tärkeä kysymys suunnittelua tehtäessä on, miten määrittelyssä olevat vaatimukset saadaan järjestelmään. Suunnittelussa pitää ottaa huomioon monta eri asiaa, esimerkiksi järjestelmän toiminnallisuus ja rakenne, käyttöliittymän eri elementit ja visuaalinen suunnittelu.

Toiminnallisuutta voi suunnitella monella eri tapaa, esimerkiksi muuttamalla toimintatarinat käyttötarinoiksi tai käyttämällä kuvatarinoita, vuokaavioita tai miellekarttoja. Toimintatarinoilla kuvataan, miten käyttäjät käyttävät vanhaa järjestelmää, ja käyttöta-

rinoilla tarkoitetaan kuvausta siitä, miten järjestelmän käyttäjät toimivat ja käyttävät uutta järjestelmää. Käyttötarinoita voidaan hyödyntää suunniteltaessa informaatioarkkitehtuuria tai yksittäistä työprosessia. Käyttötarinoiden huono puoli on visuaalisen esityksen puute. Niiden lisänä voi käyttää esimerkiksi kuvatarinoita tai vuokaavioita. Tärkeintä kuitenkin on, että suunnitelman perusajatus tulee esille ja on ymmärrettävä. (5, s. 171–172, 175.)

Kuvatarinat ovat myös hyvä tapa suunnitella järjestelmän toiminnallisuutta. Niillä kuvataan järjestelmän käyttöä kuvilla ja käyttäjän kommentteilla. Tarina kuvataan vaihe kerrallaan, ja jokaiseen vaiheeseen tulee näkymä järjestelmän käyttöliittymään sekä käyttäjän ajatukset ja reaktiot järjestelmän toiminnallisuudesta. Vuokaavio on helppo ja hyödyllinen tapa suunnitella ja kuvata järjestelmän toiminnallisuutta. Kaavio kuvaa algoritmia tai prosessia erilaisilla laatikoilla ja nuolilla. Tieto kuvataan laatikoissa, ja tiedon kulku ja yhteys kuvataan nuolilla. Vuokaavioilla voi kuvata hyvin sekä yksinkertaisia että kompleksisia prosesseja. (5, s. 178–179.) Kuvassa 9 on kuvattu yksi uuden järjestelmän perustoiminnallisuuksista eli uuden tapauksen luominen.



Kuva 9: Uuden tapauksen luominen tapauksenhallintajärjestelmässä.

Työssä tutkittavan järjestelmän toiminnallisuutta ei juurikaan suunniteltu, koska kyseessä on jo valmis järjestelmä, johon vain lisätään osia. Asiakaskäyttöliittymän puolella tullaan rajaamaan toiminnallisuuksia käyttöoikeuksilla, koska asiakkailla ei ole tarvetta nähdä kaikkea, mitä järjestelmässä on. On myös erittäin tärkeää, että asiakkaat näkevät vain omat tapauksensa eivätkä missään tapauksessa muiden asiakkaiden tapauksia.

4.2 Rakenne

Ennen kuin projektissa aloitetaan yksityiskohtainen suunnittelu ja teko, on hyvä saada kokonaiskuva järjestelmästä ja sen rakenteesta. Informaatioarkkitehtuurin suunnittelulla tarkoitetaan rakenteen ja navigoinnin suunnittelua, ja se on ensimmäinen kohta

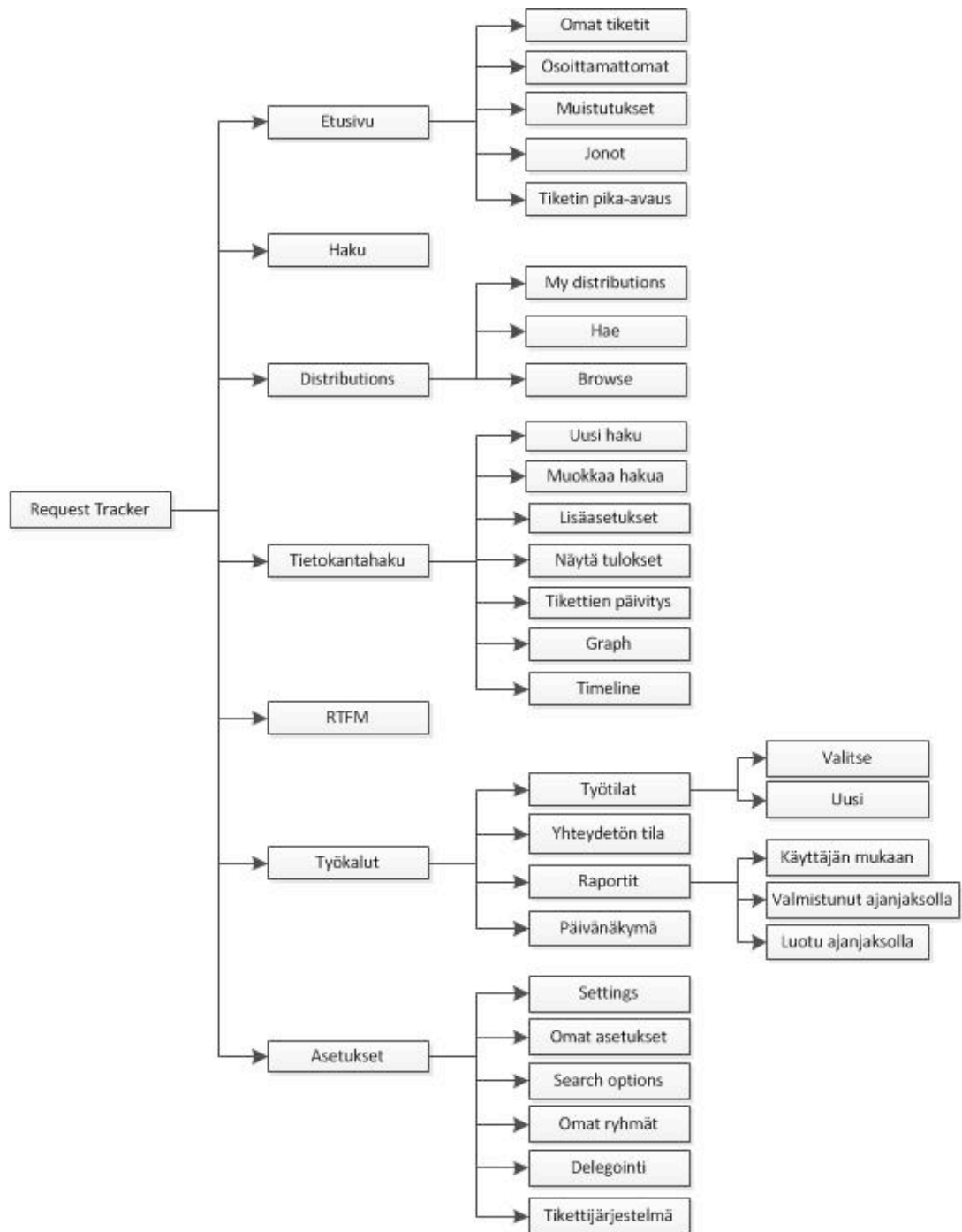
suunnittelutyössä. Informaatioarkkitehtuurin tavoitteena on helpottaa informaation selvittämistä ja kontrollointia. Järjestelmän informaatioarkkitehtuuria suunniteltaessa täytyy ottaa huomioon käyttäjien tarpeet, sisällön asettamat rajoitukset ja asioiden käsitteellinen yhteys. Kun informaatioarkkitehtuuri on onnistunut, käyttäjällä on koko ajan tunne, että hän lähenee etsimäänsä tietoa. Se myös auttaa käyttäjää hahmottamaan, missä osassa järjestelmää hän on ja miten hän on sinne päässyt. Hyvä informaatioarkkitehtuuri on myös helppo ylläpitää ja tukee käyttäjän informaation omaksuttavuutta. (5, s. 183–184.)

Järjestelmien suunnittelussa käytetään usein hierarkkista informaatioarkkitehtuuria. Sen rakenne kuvaa sujuvasti ihmisen tapaa lajitella ja organisoida asioita ala- ja yläkäsitteisiin. Hierarkkisen rakenteen omaavan järjestelmän on havaittu olevan parhaiten ymmärrettävä, ja se myös tukee tyypillisiä etenemistapoja järjestelmässä. Kun tietoa kertyy järjestelmään paljon, sen on oltava rakenteeltaan selvää, jotta käyttäjät löytävät etsimänsä tiedon. (5, s. 184–185.)

Muita rakennemalleja kuin hierarkiaan perustuva ovat esimerkiksi dynaamisiin linkkeihin perustuva rakenne, tietokannan rakenteeseen perustuva informaatorakenne tai hypertekstirakenne. Dynaamisiin linkkeihin perustuvalla rakenteella tarkoitetaan sivustoa, jonka etusivulla on vakioaiheiset nostot ja niissä kunkin alueen dynaamisia linkkiotsikoita. Tällainen rakenne on erinomainen, kun sivuston sisältöä päivitetään usein. Hyvä esimerkki tällaisista sivuista ovat iltapäivälehtien WWW-sivut. Tietokannan rakenteeseen perustuva informaatorakenne nimensä mukaan perustuu tietokannan rakenteeseen ja on harvoin onnistunut ratkaisu, koska tietokannan rakenne saattaa olla sekava ja raskas. Hypertekstirakennetta suositellaan käytettävän yhdessä hierarkkisen informaatorakenteen kanssa. Siinä integroidaan yksittäisiä informaatioelementtejä tai niiden joukkoja toisiinsa linkein. (5, s. 187–188.)

Työssä tutkittavan järjestelmän rakenne on yksinkertainen, ja siinä on käytetty hierarkkista informaatioarkkitehtuuria. Siinä on seitsemän elementtiä päänavigaatoriossa, ja niistä viidellä on alaelementtejä. Yhdellä alaelementeistä on vielä alaelementti. Välillä käyttäjän on vaikea tietää, missä kohtaa järjestelmää on menossa, koska arkkitehtuuria ei ole suunniteltu tarpeeksi tarkasti. Koska järjestelmään kertyy paljon tietoa, olisi tär-

keää, että käyttäjä löytää tiedon nopeasti ja helposti. Kuvassa 10 on nähtävissä järjestelmän rakenne.



Kuva 10: Tapauksenhallintajärjestelmän rakenne.

4.3 Järjestelmän käyttöliittymä

4.3.1 Käyttöliittymän elementit

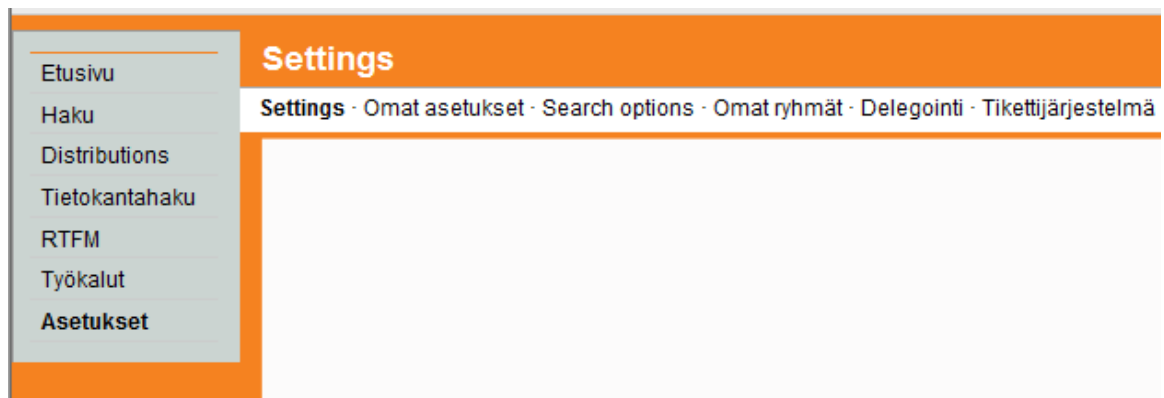
Käyttöliittymällä tarkoitetaan tapaa, jolla tietosisältö on käytettävissä. Koska käyttöliittymä on useasti näköaistiin perustuva, sillä useimmiten tarkoitetaan visuaalista näkymää, joka näkyy järjestelmän tai sovelluksen käyttäjälle. Se on myös se osa sovelluksesta, jonka avulla käyttäjä ja sovellus ovat vuorovaikutuksessa keskenään. Käyttöliittymä kattaa myös esimerkiksi tavat siirtyä toiselle sivulle tai etsiä tietoa. (7, s. 354.)

Käyttöliittymä koostuu useista elementeistä, muun muassa navigointielementeistä, hakutoiminnoista ja verkkolomakkeista. Käyttöliittymän elementit vaihtelevat käyttöliittymästä riippuen. Yksi elementti, joka on lähes kaikissa käyttöliittymissä, on navigointielementti. Ilman sitä käyttöliittymä rajoittuu vaan yhteen sivuun eikä siitä voi siirtyä mihinkään. Navigointielementit auttavat käyttäjää siirtymään sivulta toiselle ja näkemään missä hän on järjestelmässä ja mihin hän sieltä pääsee. Navigointielementit on hyvä pitää visuaalisesti erottuvina, mutta ei liian dominoivina. Navigoinnin tärkeyttä ei pidä vähätellä. Navigaation voisi rinnastaa opasteisiin ja tienviittoihin, jotka auttavat käyttäjää löytämään etsimänsä. (5, s. 215; 8, s. 154.)

Navigointivalikoita on erilaisia. Koko palvelun tai järjestelmän tasoisella eli globaalilla navigointijärjestelmällä tarkoitetaan koko sivuston päävalikkoa, josta käyttäjä pääsee järjestelmän eri osiin. Päävalikko on kaikilla järjestelmän sivuilla samanlainen riippumatta siitä, missä kohtaa järjestelmää käyttäjä sijaitsee. Päävalikko useimmiten sijaitsee joko vaakavalikkona sivun ylälaidassa tai sivuvalikkona sivun vasemmassa laidassa. Toisenlainen navigointijärjestelmä on paikallinen eli lokaali navigointijärjestelmä. Sillä tarkoitetaan navigaatiota, joka on erilainen jokaisessa sivuston osassa. Paikallinen navigaatio sijaitsee yleensä visuaalisesti päävalikon alla. Useimmiten paikallinen valikko on joko sivun vasemmassa laidassa oleva pystyvalikko tai päävalikon alla oleva vaakavalikko. Paikallinen valikko voi myös olla päävalikosta klikkaamalla avautuva pudotusvalikko. (5, s. 216–217.)

Työssä tutkittavan tapauksen hallintajärjestelmän pää- eli globaalivalikko on sivujen vasemmassa laidassa harmaalla pohjalla oleva pystyvalikko, ja paikallinen eli lokaaliva-

likko aukeaa sivun yläreunaan. Kaikissa päävalikon sivuissa ei ole käytössä lokaalia valikkoa. Kuvassa 11 on esitetty järjestelmän pää- ja lokaalivalikko.



Kuva 11: Uuden tapauksenhallintajärjestelmän pää- ja lokaalivalikko.

Käyttäjillä on erilaisia tottumuksia hakea tietoa järjestelmistä. Hakutoimintoa voi käyttää, kun tietää, mitä on etsimässä. Järjestelmissä on usein erilaisia hakutoimintoja. Monessa järjestelmässä on oikeassa yläkulmassa hakutoiminnallisuus, jonka avulla voi helposti etsiä tietoa. Tätä sanotaan yksinkertaiseksi hauksi. Se etsii sivuston sisältöä, joka täyttää annetut ehdot. Monissa järjestelmissä on myös laajennettu haku, jossa käytössä on enemmän hakukriteereitä ja rajausmahdollisuuksia. Työssä tutkittavassa järjestelmässä on olemassa pikahaku järjestelmän oikeassa yläkulmassa. Sen avulla voi etsiä tapauksia kaikista jonoista, joihin käyttäjällä on oikeudet, hakemalla tapauksen ID-numerolla tai otsikolla. Järjestelmässä on myös mahdollisuus tehdä laajennettu tietokantahaku, jossa hakukriteerejä on enemmän. Hakuun voi ottaa mukaan tai rajata hakukriteerejä oman maun mukaan muun muassa seuraavista vaihtoehtoista:

- ID-numero (on yhtä kuin, pienempi kuin, suurempi kuin)
- otsikko (täsmää, ei täsmää)
- asiakasjono (on, ei ole)
- tapauksen status (on, ei ole)
- tapauksen omistaja (on, ei ole)
- asiakaskontaktin sähköposti (täsmää, ei täsmää)
- luontipäivämäärä (ennen, jälkeen)
- prioriteetti (enemmän kuin, vähemmän kuin, on yhtä kuin)
- tapauksen tyyppi (täsmää, ei täsmää).

Listaa voisi jatkaa, mutta tämä jo kuvastaa, kuinka laajasta tietokantahausta on kysymys. (8, s. 221–222.)

Verkkolomakkeet ovat yleistyneet järjestelmissä ja sovelluksissa. Lomakkeet ovat hyödyllisiä, kun halutaan antaa tietoa järjestelmälle tai järjestelmän takana olevalla yritykselle. Tiedot voivat muun muassa olla palautteen antamista, tilaus tai yhteydenotto-
pyyntö. Useimmiten käyttäjän ja järjestelmän välisessä vuorovaikutustilanteessa käyttäjä etsii, valitsee ja lukee tietoja järjestelmän tietokannasta ja pienempi käyttäjänryhmä voi myös lisätä, muokata tai poistaa tietokannan tietoja. Tutkittavassa tapauksessa hallintajärjestelmässä esiintyy molempia käyttäjiä. Asiakaskäyttäjät useimmiten etsivät ja lukevat tietokantaan tallennettuja tapauksia, mutta myös luovat itse uusia tapauksia, jotka tallentuvat tietokantaan. Asiakaspalvelijat tekevät enemmän muutoksia tai poistavat tapauksia tietokannasta. Nämä mahdollisuudet riippuvat myös käyttöoikeuksista. Asiakkaalle ei haluta antaa oikeuksia tehdä mitä vain järjestelmässä eikä varsinkaan tietokantaan, joka on järjestelmän perustana. (5, s. 222.)

4.3.2 Visuaalinen suunnittelu

Visuaalinen suunnittelu on tärkeä osa järjestelmää ja sen suunnittelua, mikä on opittu ajan myötä. Kun Internet teki tuloaan, sivustojen ja järjestelmien visuaalinen ulkoasu ei ollut kovin imarteleva. Pahimmassa tapauksessa ne koostuivat harmaasta pohjaväristä, jonka päällä oli mustaa tekstiä. Luettavuus tällä yhdistelmällä ei ollut paras mahdollinen. Ajan myötä, kun selaimet kehittyivät, myös visuaalinen suunnittelu otti ison harppauksen eteenpäin, minkä käyttäjätkin harmikseen huomasivat. Nimittäin grafiikka, värit, animaatiot ja erilaiset tekstityypit pompahtivat niin voimakkaasti käyttäjän eteen, ettei niitä voinut olla huomaamatta. Nykyään niitä osataan jo käyttää maltillisemmin. (7, s. 355.)

Järjestelmän visuaalinen ulkoasu on keskeinen osa käytettävyyttä ja näin ollen viestii käyttäjälle kahta viestiä. Ensimmäinen viesti on järjestelmän sisällön esittäminen: ulkonäön pitää auttaa käyttäjää huomaamaan, jäsentämään ja ymmärtämään asiat. Toinen viesti on järjestelmän kokonaisilmeeseen liittyvä: minkälainen on järjestelmän kokonaisilme, tunnelma ja persoonallisuus. (8, s. 90; 5, s. 242.)

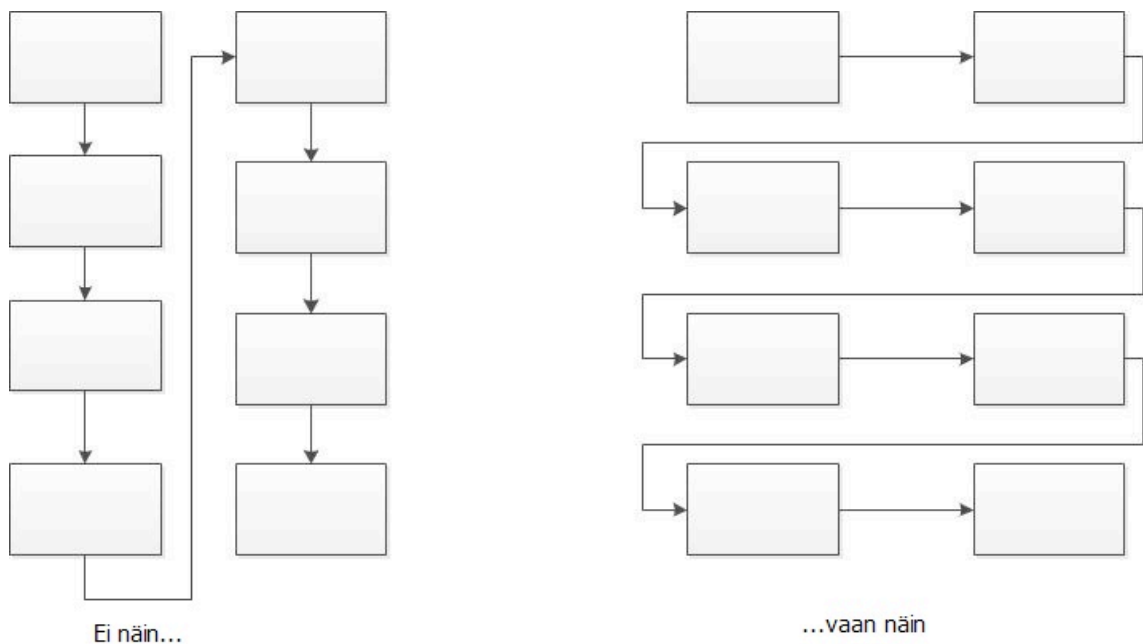
Visuaalinen suunnittelu aloitetaan hyvissä ajoin projektin alussa, kun informaatioarkkitehtuuri on valmis, ja sitä tehdään koko ajan projektin rinnalla. Visuaalista ulkoasua

olisi hyvä testata projektin edetessä. Visuaalinen suunnittelu voidaan aloittaa tyhjästä tai uudistaa vanha ilme. Heti kun elementit on suunniteltu ja pantu paikoilleen sivuille, kootaan sivut ilman sisältöä. Sivujen visuaalinen suunnittelu ja toteutus voi viedä paljonkin aikaa, joten viimeisenä sivuille upotetaan sisältö. (5, s. 242–243; 8, s. 91.)

Visuaalisen suunnittelun tavoitteina voidaan pitää esteettisyyttä, käytettävyyttä, brändin esille tuomista, persoonallisuutta ja hyvää kokonaisilmettä. Esteettisyydellä tarkoitetaan arkikielessä kauneutta ja taidetta, ja sillä on suuri merkitys visuaalisessa suunnittelussa, koska usein käyttäjät pitävät esteettisiä järjestelmiä laadukkaampina ja luotettavampina kuin muita järjestelmiä. Vaikka esteettisyys ja käytettävyys ovat yksittäisiä ominaisuuksia, on niillä kuitenkin yhteys. Käytävyydellä tarkoitetaan järjestelmän ja käyttäjän välistä käyttökokemusta. Brändillä tarkoitetaan yrityksen arvoja ja toimintatapaa. Järjestelmien tulisi mukailla yrityksen brändiä, koska ne omalla tavalla mainostavat ja tuovat esille yritystä. Nykypäivänä selainpohjaisia järjestelmiä toteutetaan enemmän kuin koskaan, joten persoonallisuus on tärkeää. Persoonallinen järjestelmä tai palvelu herättää käyttäjän kiinnostuksen yleisimmin visuaalisella ilmeellä, muun muassa sijoittelulla, väreillä, elementtien tyylillä. Persoonallisella järjestelmällä on myös hyvä kokonaisilme. Se syntyy muun muassa väreistä ja yhtenäisistä muodoista ja typografiasta. (5, s. 249–251.)

Visuaaliseen suunnitteluun on erilaisia keinoja, muun muassa visuaalinen hierarkia, katseen ohjaus, ryhmittely, tyhjä tila, sommittelu, tasapaino, värit, typografia ja kuvat. Visuaalisella hierarkialla tarkoitetaan tosielämässä olevaa hierarkkista suhdetta, joka sivuille tai järjestelmään tulevilla asioilla on. Asioiden kuulumisen ja liittymisen toisiinsa tulisi näkyä esimerkiksi asemoinnissa. Katseen ohjaus on tärkeässä roolissa, varsinkin jos käyttäjä kokeilee järjestelmää ensimmäistä kertaa. Visuaalisella suunnittelulla autetaan käyttäjää viemään katse oikeaan paikkaan ja sitä kautta löytämään etsimänsä. Ryhmittelyn tarkoituksena on helpottaa käyttäjien hahmottamista ja muistamista. Ryhmittely myös helpottaa ja nopeuttaa silmäilyä. Tyhjä tila elementtien ympärillä on tärkeä, koska sillä korostetaan niiden tärkeyttä. Sommittelulla haetaan tasapainoista asettelua. Sommittelu on osattava tehdä hyvin, koska epäonnistunut sommittelu vie koko idean järjestelmästä ja sivustolta. Sommitteluun voi yhdistää myös lukusuunnan mukaisen järjestyksen ja siinä pysymisen. Kuva 12 kuvastaa, miten lukusuunnasta poikkeaminen vaikeuttaa järjestelmän käyttöä. Värit, typografia ja kuvat ovat vahva

osa koko visuaalista ilmettä, joten niiden suhteen pitää olla kunnossa. (5, s. 251–255; 8, s. 91–92).



Kuva 12: Lukusuunnan havainnollistaminen (8).

Työssä tutkittavan järjestelmän visuaalista ilmettä ei lähdetty muuttamaan projektin aikana, vaikka parannettavaa löytyisikin. Järjestelmän käytettävyys on hyvä vaikkakin vähän epäselvä. Välillä on vaikea tietää, mistä haettu tieto löytyy. Järjestelmän ulkoasu ei tuo yrityksen brändiä esille muuten kuin värimaailmallisesti. Yrityksen ilme ja logo päivitettiin syksyn 2010 aikana, joten ne olisi hyvä päivittää myös järjestelmään. Järjestelmästä puuttuu myös persoonallisuus. Se on ulkoasultaan hyvin yksinkertainen, eikä siinä ole mitään, mikä erottaisi sen muista tarjolla olevista järjestelmistä. Järjestelmän ulkoasu on ryhmittelyltään selkeä, ja tyhjää tilaa löytyy elementtien ympäriltä, mikä tosin välillä hankaloittaa järjestelmä selailua. Elementtejä on niin runsaasti ja kaikkien välissä on tyhjää tilaa, että se tekee järjestelmän sivuista erittäin pitkiä ja hankalia selattavaksi. Sommittelu voisi olla selkeämpi. Tällä hetkellä esimerkiksi järjestelmän etusivulla on paljon elementtejä, joita suurin osa käyttäjistä ei tarvitse. Olisi selkeämpää, jos tarpeettomat elementit voisi piilottaa ja jättää vain tarpeelliset näkyviin. Järjestelmän nykyisessä versiossa on käytössä yleisvärinä kirkkaan oranssi, joka on erittäin kirkas ja hieman rauhaton. Värityys voisi olla kokonaisuudessaan käyttäjäystävällisempi ja rauhallisempi. Uudemmassa versiossa yleisväri on tummansininen, joka on jo paljon

rauhallisempi, selkeämpi ja auttaa hahmottamaan paremmin. Järjestelmän typografia on selkeä ja helppolukuinen. Järjestelmässä tosin on sekaisin suomea ja englantia, mikä pitäisi yhtenäistää. Kuvassa 13 on nähtävissä uuden järjestelmän hakusivun ulkoasu.

Kuva 13: Uuden järjestelmän hakusivu.

5 Tapauksenhallintajärjestelmästä puuttuvat ominaisuudet

5.1 Vasteajat

Vasteajoilla tarkoitetaan tutkimis- ja ratkaisuaikaa virhetilanteessa. Vasteajat määritellään tukipalvelusopimuksessa, joka solmitaan toimittajan ja asiakkaan välille. Vasteajat voidaan jakaa kolmeen osaan: tapahtuman tutkiminen ja vahvistaminen (tapahtumanhallinta), väliaikainen korjaus tai ohjeet, joilla tapahtuma vältetään (tapahtumanhallinta) ja lopullinen korjaus tai ohjeet, joilla virhe vältetään (ongelmanhallinta). Tukipalvelusopimuksessa määritellään tapahtumanhallinnan ja ongelmanhallinnan tavoite. Tapahtumahallinnan tavoite on palauttaa palvelun normaali toiminta mahdollisimman nopeasti ja mahdollisimman vähäisillä häiriöillä liiketoiminnalle. Ongelmanhallinnan tavoitteeksi on asetettu vähentää virheiden aiheuttamien tapahtumien ja häiriöiden vaikutuksia palvelun käyttöön ja virheiden aiheuttamien tapahtumien, häiriöiden ja virheiden uusiutuminen.

Vasteajat ovat asiakaskohtaisia. Joillakin asiakkailla ei ole vasteaikoja ollenkaan. Vasteajat ovat riippuvaisia myös asiakkaan projektista, tapauksen prioriteetista ja tyypistä. Asiakkaan tapaukset luokitellaan kiireellisyyden mukaan neljään prioriteettiin (A, B, C, D). Taulukossa 1 ovat nähtävissä kaikki neljä prioriteettia ja tapahtuman kuvaus, joka vastaa kyseistä prioriteettia. Tapauksella on kolme eri tyyppiä, joka vaikuttavat vasteaikaan. Ne ovat virhetilanne (incident), palvelupyyntö, joka voi olla kysymys tai pieni muutos (service request), ja ongelmatilanne (problem). Myös asiakkaan palveluaika vaikuttaa vasteaikojen laskentaan. Asiakkailla on kahta erilaista palveluaikaa: laajempi klo 8-18 tai suppeampi klo 9–17. Vasteajat lasketaan näiden palveluaikojen puitteissa. Myöhemmin työssä käydään tarkemmin läpi laskentaa.

Taulukko 1. Esimerkkiasiakkaan prioriteettitaulukko.

Luokka	Prioriteetti järjestelmässä	Tapahtuman kuvaus
A	korkein	Ohjelmiston tapahtuma/ongelma estää asiakasta täysin käyttämästä ohjelmistoa, ja on näin ollen asiakkaan liiketoiminnan kannalta kriittinen.
B	korkea	Tapahtuma/ongelma estää asiakasta käyttämästä vähintään yhtä ohjelmiston päätoimintoa, mutta ei estä asiakasta käyttämästä koko ohjelmistoa eikä ole liiketoiminnan kannalta kriittinen.
C	normaali	Tapahtuma/ongelma estää asiakasta käyttämästä vähintään yhtä ohjelmiston perustoimintoa, jota ei voida pitää päätoimintona. Tapahtuma/ongelma ei estä ohjelmiston peruskäyttämistä.
D	alhainen, alhaisin	Ohjelmiston tapahtumalla/ongelmalla ei ole asiakkaan liiketoiminnan kannalta merkitystä, eikä virhe kuulu mihinkään yllä olevaan luokkaan.

Tapauksen status vaikuttaa vasteaikojen kulkuun. Vasteaika alkaa kulkea, kun asiakas kirjaa tapauksen järjestelmään. Ensimmäinen kohta laskennassa on tapauksen vahvis-

taminen, mikä tarkoittaa asiakaspalvelun kuittaamista, että tapaus on vastaanotettu. Vasteaika kulkee eteenpäin koko ajan, jos tapaus on toimittajan teknisellä asiantuntijalla työn alla. Kun asiakasta pyydetään testaamaan tai kysytään asiakkaalta lisätietoja, status vaihdetaan odottaa-tilaan ja vasteajan kuluminen pysähtyy. Kun asiakas vastaa tarvittavat tiedot ja tekninen asiantuntija pystyy taas työstämään tapauksta, vasteaika kuluu taas. Aika, jonka tapaus oli odottaa-tilassa, lisätään kokonaisaikaan. Taulukko 2 on esimerkkiasiakkaan vasteaikataulukko prioriteetin mukaan.

Taulukko 2. Esimerkkiasiakkaan vasteaikataulukko.

Kiireellisyysluokka	Tapahtuman vasteaika (Tapahtumanhallinta)	Väliaikainen korjaus tai ohjeet, joilla tapahtuma vältetään (Tapahtumanhallinta)	Lopullinen korjaus tai ohjeet, joilla virhe vältetään (Ongelmanhallinta)
A	8 h	16 h	40 h
B	16 h	40 h	80 h
C	24 h	80 h	160 h
D	40 h	320 h	uusi kehitysprojekti

Taulukossa 2 esimerkkiasiakkaan palveluaika on maanantaista perjantaihin arkipyhät pois lukien kello 9–17 (Suomen aikaa). Asiakas kirjaa tapauksen C-prioriteetilla järjestelmään maanantaina 22.11.2010 klo 9.15. Taulukon ensimmäisen sarakkeen mukaan C-prioriteetin tapaus pitää kuitata vastaanotetuksi asiakaspalvelussa 24 tunnin kulussa. Taulukon toisen sarakkeen mukaan teknisellä asiantuntijalla on 80 tuntia aikaa saada väliaikainen korjaus tai ohjeet valmiiksi, eli vasteaika tulisi täyteen 3.12.2010 kello 9.15. Keskiviikkona 24.11.2010 kello 12.45 asiakaspalvelu kysyy asiakkaalta lisätietoja ja laittaa tapauksen odottaa-tilaan, jolloin vasteikalaskenta pysähtyy. Asiakas vastaa torstaina 25.11.2010 kello 15.55, jolloin vasteikalaskenta käynnistyy taas. Kokonaisaikaan lisätään aika, jonka tapaus on ollut odottaa-tilassa. Tapaus oli odottaa-tilassa 11 tuntia ja 10 minuuttia palveluaikojen sisällä, joten tämä aika lisätään alkuperäiseen vasteaikaan. Lisäyksen jälkeen vasteaika ylittyisi keskiviikkona 8.12.2010 kello 12.25. Liitteessä 2 on tarkempi laskukaava yllä olevasta esimerkistä.

Nykyisessä tapauksenhallintajärjestelmässä on käytössä vasteaikapalkki, jossa näkyvät kaikki kolme eri päivämäärää. Kuvassa 14 on nähtävissä esimerkki vasteaikapalkista. Palkki on vihreä, kun vasteaika on vielä jäljellä. Kun vasteaika on jäljellä enää muutama tunti, palkki muuttuu keltaiseksi, ja kun vasteaika on ylittynyt, palkki muuttuu punaiseksi. Värien avulla on helppo seurata vasteajan kehittymistä ja kulumista. Uudessa järjestelmässä ei ole ollenkaan vasteaikoja käytössä, joten kaikki pitää kehittää ja ohjelmoida alusta asti. Määrittelyvaiheessa oli alusta asti selvää, että samanlaista palkkia ei lähdetä ohjelmoimaan, koska se olisi hankala toteuttaa uuteen järjestelmään. Parhaaksi vaihtoehdoksi todettiin päivämääräkentät, joihin tarvittavat päivämäärät tulisivat näkyviin. Tämä on selvästi huonompi vaihtoehto, koska palkki on paljon havainnollisempi kuin pelkkä päivämääräkenttä.



Kuva 14: Vanhan tapauksenhallintajärjestelmän vasteaikapalkki.

5.2 Jononäkymä

Jononäkymällä tarkoitetaan listausta kaikista avoinna olevista tapauksista, jotka ovat järjestelmässä. Asiakaspalvelun tehtävä on seurata kaikkia avoinna olevia tapauksia asiakkaasta ja statuksesta riippumatta. Vanhassa järjestelmässä jononäkymä on toteutettu erittäin järkevästi. Näkymässä näkyvät kaikki avoimet tapaukset asiakkaasta ja statuksesta riippumatta. Yksittäisestä tapauksesta näkyy ID-numero, prioriteetti, vasteaika, status, otsikko, asiakas, asiakaspalvelija, joka on ottanut tapauksen vastaan, ja kehittäjä, jolle tapaus on osoitettu. Näkymästä on helppo seurata tapausten tilaa ja etenemistä.

Uudessa järjestelmässä on tällä hetkellä jokaisella asiakkaalla oma jono, mikä tarkoittaa sitä, että kaikkien asiakkaiden avoimia tapauksia on mahdotonta seurata kerralla. Kaikkia avoimia tapauksia ei ole missään nähtävissä, joten asiakaspalvelun työtä on käytännössä mahdotonta tehdä. Jos avoimet tapaukset haluaisi käydä läpi, ne pitäisi

käydä asiakasjono kerrallaan läpi ja se veisi valtavasti aikaa. Projektin määrittelyvaiheessa oli selvää, että ilman kaikkien tapausten jononäkymää asiakaspalvelu ei voisi käyttää järjestelmää, joten vaatimuksen prioriteettia nostettiin.

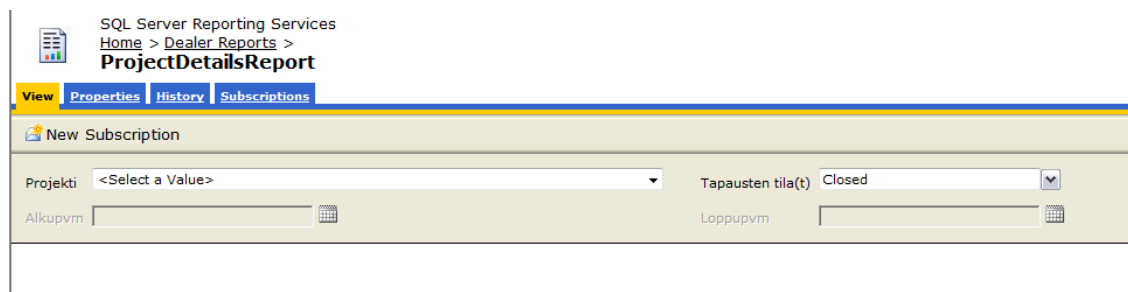
5.3 Raportointi

Raportoinnilla tarkoitetaan palvelupäälliköiden tekemiä vasteaikaraportteja. Tukipalvelusopimuksessa, joka solmitaan asiakkaan ja toimittajan välillä, määritellään palvelun sisältävän kuukausittain tai kvartaaleittain (4 raporttia/vuosi) tehtävän vakioidun seurantaraportin suljetuista tapauksista. Raportoinnin avulla asiakas saa tiedon tapauksista ja niiden vasteajoista. Raportoinnin avulla saadaan selville myös mahdolliset sanktiot esimerkiksi vasteajan ylityksestä.

Vanhasta järjestelmästä on otettu kahdenlaisia raportteja: raportti suljetuista tapauksista ja asiakaspalauteraportti. Uudesta järjestelmästä määriteltiin näiden lisäksi vielä otettavaksi raportti avoinna olevista tapauksista eri statuksilla. Suljettujen tapausten raporttiin kerätään kaikki määritettynä aikavälinä suljetut tapaukset tietyltä asiakkaalta ja tietyltä asiakkaan projektilta. Vastaavasti avoinna olevien tapausten raporttiin kerätään kaikki määritettynä aikavälinä avoimet tapaukset tietyltä asiakkaalta ja tietyltä asiakkaan projektilta. Suljettujen ja avoinna olevien tapauksien raportissa näkyy tapausten lukumäärä, luettelo tapauksista prioriteeteittain, luettelo tapausten keskimääräisistä vasteajoista prioriteeteittain, tapausten keskimääräinen ratkaisuaika, luettelo prioriteettijärjestyksessä tapauksista, jotka ovat ylittäneet ratkaisuvasteajan, sekä ylittynyt aika. Asiakaspalauteraporttiin määritellään aloitus- ja päättymispäivämäärä sekä projektit, joista halutaan palaute. Raportissa näytetään kaikki tapaukset ja muutokset, jotka on suljettu ja annettu palautetta määritettynä aikavälinä.

Raportointi toteutetaan Microsoftin SQL Server Reporting Services -ohjelmalla. Ohjelmaa käytetään Internetissä olevan käyttöliittymän kautta. Se tarjoaa täyden valikoiman käyttövalmiita työkaluja ja palveluita, jotka auttavat luomaan, käyttämään ja käsittelemään erilaisia raportteja organisaatiossa. Ohjelmassa on myös ohjelmointiominaisuus, jonka avulla raporttitoiminnallisuuksia voi laajentaa ja muuttaa yrityksen tarpeita vastaaviksi. Reporting Services on palvelinperäinen raportointialusta, joka tarjoaa kokonaisvaltaisen raportointitoiminnallisuuden vaihteleville tietolähteille. Se sisältää valmiin sar-

jan työkaluja raporttien luomiseen, käsittelemiseen ja toimittamiseen. Työkalut toimivat Microsoftin Visual Studio -ympäristössä, ja ne on yhdistetty SQL Server -työkaluihin ja komponentteihin. (10.) Kuvassa 15 on nähtävissä SQL Server Reporting Services -ohjelman käyttöliittymä.



Kuva 15: SQL Server Reporting Services-ohjelman käyttöliittymä.

6 Projektin kulku

6.1 Projektin eteneminen ja ketterät toimintatavat

Kun suunnitteluvaihe on valmis, on edessä projektin käynnistäminen eli toteutusvaihe. Vaikka kyseessä on ehkä itsestään selvä asia, on syytä tehdä selväksi koko projekti-ryhmälle, milloin projekti käynnistetään konkreettisesti. Yleisin tapa tähän on järjestää projektin johtoryhmän kokous, jossa käydään läpi ja hyväksytään projektisuunnitelma. Kokouksessa tehdään myös virallinen päätös projektin käynnistämisestä. (2, s. 58.)

Projektin toteutusvaihe voidaan jakaa kahteen osaan, ohjaus- ja suoritustehtäviin. Ohjaustehtävillä tarkoitetaan projektin seuranta ja ohjausta. Työssä tutkitavan järjestelmän kehitysprojektin seurantaan osallistui kehitystiimi, johon kuului minun lisäksi projektipäällikkö ja tekninen asiantuntija. Projektin alkuperäinen idea tuli johtoryhmästä, joten mielestäni johtoryhmän jäsenten olisi ollut hyvä osallistua projektin seurantaan ja ohjaukseen enemmän. Näin projektille olisi mahdollisesti saatu enemmän painoarvoa ja asiat olisivat tapahtuneet nopeammin. (2, s. 58.)

Suoritustehtävien tuloksena syntyy projektin varsinainen tuote, eli uusi tapauksenhallintajärjestelmä. Ennen kuin päästään valmiiseen tuotteeseen asti, on toteutusvaiheessa monta kohtaa päätettävänä. Ensimmäinen on toteutusvälineen tai ohjelmointikielen valinta. Tähän valintaan vaikuttavat muun muassa seuraavat asiat:

- sovellusalue
- käytetyt menetelmät ja ohjelmistotuotannon mallit
- tehokkuusvaatimukset
- toteutus- ja käyttöympäristöt.

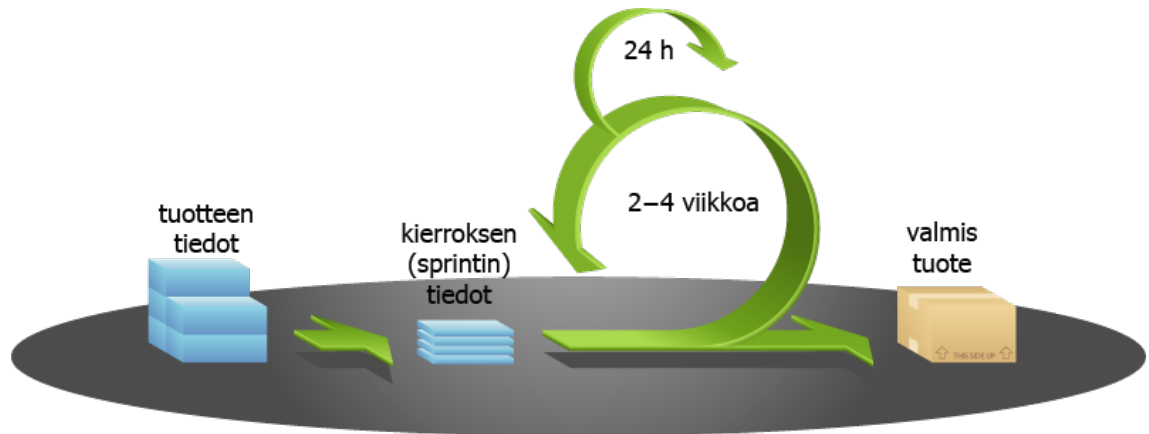
Työssä tutkittavassa järjestelmän toteutuksessa ei ollut tarvetta valita ohjelmointikieltä, koska kyseessä on kehitysprojekti jo olemassa olevaan järjestelmään. Järjestelmä on toteutettu Perl-kielellä, ja sillä myös muutokset tehdään. (2, s. 34.)

Nykyajan projektitoiminta on ajautunut melkein umpikujaan markkinoiden asettamien vaatimusten kanssa. Koko ajan halutaan korkeampilaatuisia järjestelmiä, joiden kulujen pitäisi olla mahdollisimman alhaiset. Sama ilmiö on huomattavissa työssä tutkittavassa järjestelmän kehittämisprojektissa. Markkinoilla olisi tarjolla monia eri järjestelmävaihtoehtoja, jotka vastaisivat vaatimuksia jo valmiiksi, mutta kustannuksissa halutaan säästää ja kehittää järjestelmä itse. Tätä ilmiötä tukemaan ja helpottamaan on kehitetty erilaisia ketterän projektitoiminnan toimintatapoja. (11, s. 20.)

Ketterät toimintatavat ovat yleistyneet projektitoiminnassa viime vuosien aikana. Niitä on monia erilaisia projektista riippuen. Tunnetuimpia toimintatapoja ovat

- Agile Modeling
- Dynamic Systems Development Method
- Extreme Programming
- Scrum.

Agile Modeling -menetelmällä tarkoitetaan kokoelmaa ohjelmiston mallintamisen arvoja, periaatteita ja tapoja, joita voidaan soveltaa ohjelmistokehityksessä joustavammin kuin perinteisiä mallinnusmenetelmiä. Dynamics Systems Development Method -menetelmällä tarkoitetaan lähestymistapaa, joka käyttää prototyyppiin perustuvia toistuvia menetelmiä. Koko projektin elinkaaren aikana käyttäjät ovat mukana projektissa. Extrame Programming -menetelmän toimivuus ja suosio perustuvat viiteen arvoon: viestintä, palaute, yksinkertaisuus, rohkeus ja kunnioitus. Scrum-menetelmää käytettäessä tavoitteena on saada tuote hiljalleen valmiiksi useiden kierrosten aikana. Kierrosta kutsutaan sprintiksi, joka kestää 2–4 viikkoa. Sprintin aikana projektia viedään eteenpäin ja tuotetta kehitetään ennalta sovitun määritelmän mukaan. (12, s. 11–25.) Kuvassa 16 on nähtävissä Scrumin toimintamenetelmä.



Kuva 16: Scrumin toimintamenetelmä (13).

Kun ketteriä menetelmiä verrataan työssä tutkittavaan tapauksenhallintajärjestelmän kehitysprojektiin, on havaittavissa, että yksi menetelmä ei suoraan vastaa projektin toimintamenetelmää. Agile Modeling -menetelmä keskittyy suunnitteluun, mikä yhdistää sen kehitysprojektiin. Projektin käyttöönoton venyessä kiinnitettiin huomiota suunnitteluun ja sen tärkeyteen. Projektin alkaessa tehtiin vaatimusmäärittely nopeasti, mutta aikataulun viivästyttyä siihen ja muuhun järjestelmän suunnitteluun käytettiin enemmän aikaa. Dynamics System Development Method -menetelmä keskittyy tuottamaan vaaditut toiminnallisuudet nopeasti järjestelmään ylittämättä projektin aikataulua ja budjettia. Tämä olisi ollut erittäin hyödyllinen menetelmä kehitysprojektissa, koska nyt aikataulu ylittyi reilusti alkuperäisestä. Verrattaessa Extreme Programming -menetelmää kehitysprojektiin, niissä on havaittavissa yhteinen piirre. Menetelmän tavoitteena on parantaa ohjelmiston laatua ja vastata asiakkaan muuttuviin vaatimuksiin. Projektissa vaatimukset eivät muuttuneet, mutta niitä tuli lisää ja niitä tarkennettiin koko ajan, jotta järjestelmästä saataisiin mahdollisimman tarpeita vastaava. Scrum-menetelmä keskittyy projektin vaiheistamiseen, ja se on nähtävissä myös kehitysprojektissa. Scrumin tärkeä ominaisuus on viestintä, johon kuuluvat myös päivittäiset pienet kokoukset. Näissä kokouksissa on tärkeä käydä läpi tilanne niin, että kaikki ymmärtävät, mitä on tehty ja miten nyt edetään. Näitä kokouksia ei pidetty kehitysprojektissa, koska sen eteneminen oli hyvin hidasta. Jos aikataulu olisi pitänyt alkuperäisen suunnitelman mukaan, olisi varmasti tällaisille kokouksille ollut tarvetta.

Dokumentaatio ja siihen suhtautuminen vaihtelevat verrattaessa niin sanottua vanhan ajan projektitoimintaa ja ketteriä menetelmiä. Ennen on ehkä uskottu määrän korvaavan laadun dokumentaatioissa. Määrittely-, suunnittelu- ja ylläpitodokumentaatiota

saattoi kertyä useita erilaisia, mutta laadultaan ne eivät täyttäneet vaatimuksia. Ketterien menetelmien toimintatavoissa ajatusmaailma on muuttunut täysin. Laadun ajatellaan olevan tärkeämpi ominaisuus. Laatu voidaan tietenkin määritellä monella eri tavalla, mutta jos dokumentaatiossa on kuvattu, mihin sitä käytetään, kenelle se on tarkoitettu ja mitä sillä yritetään saada aikaan, puhutaan jo suhteellisen laadukkaasta dokumentaatiosta. (11, s. 251–252.)

Vaikka projektissa käytettäisiin jotakin ketterän projektitoiminnan menetelmää, on aina olemassa riski projektin epäonnistumisesta. Suurimmat syyt projektin epäonnistumiseen ovat viestinnän ja kommunikoinnin puuttuminen, ominaisuuksien lisääminen jälkeenkäin, ylioptimistinen aikataulu ja tuotannossa esiintyvät ongelmat. Ominaisuuksien lisäämiseen jälkeenkäin voi olla kaksi syytä: jos järjestelmään halutaan kokonaan uusia ominaisuuksia tai jo määritellyt ominaisuudet eivät toimi kuten oli suunniteltu ja niitä pitää työstää lisää. Ominaisuuksien lisääminen ei tuota ongelmia, jos ne eivät vaikuta budjettiin tai aikatauluun. Muutoksien pitää tapahtua vähällä vaivalla ja mahdollisimman pienellä muutoksella budjettiin, jotta johto voi ne hyväksyä. Aikataulun ja työomärrän arviointiin vaikuttavat monet asiat. Vaikka aikataulu olisi mietitty monesta eri näkökulmasta, voi projektin aikana silti tulla ongelmia, jotka hidastuttavat työtä. Aikataulun arvioinnissa pitää huomioida muun muassa teknisen asiantuntijan taitotaso, työkalujen toiminta ja toistokierroksien määrä. Tuotannon haasteita on maksimoida tehokkuus, minimoida hukkatyötunnit ja luoda odotustenmukainen tuote. (11, s. 17–19.)

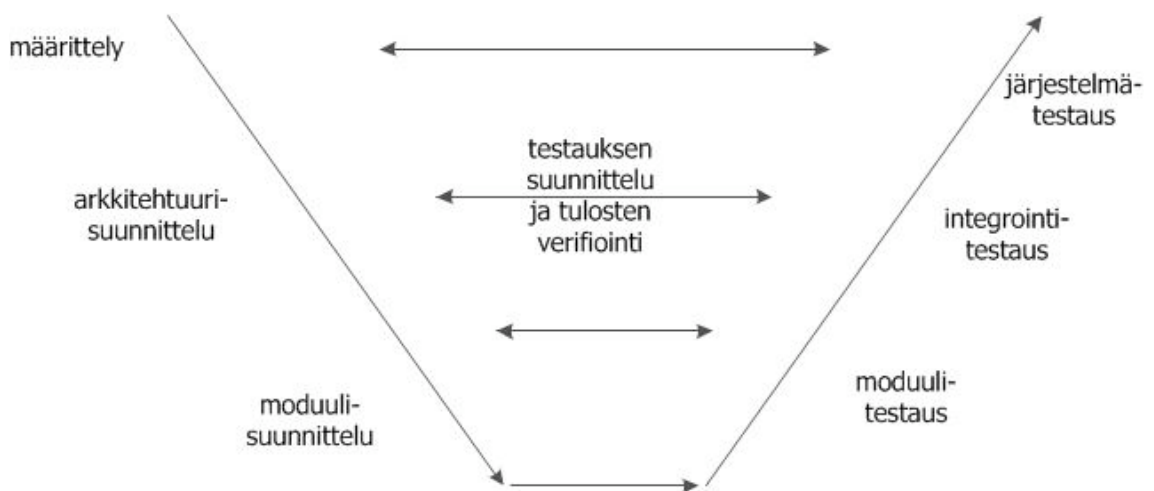
Työssä tutkittavan kehitysprojektin epäonnistumisen päällimmäinen syy on ylioptimistisen aikataulu. Teknisen asiantuntijan taitotaso arvioitiin oikein, mutta asiakastöiden määrä yllätti. Projektin alkaessa oli arvioitu asiantuntijan pystyvän tekemään järjestelmän kehittämistä 2–3 päivänä viikossa, mutta yhtäkkiä hän olikin täysin työllistetty asiakasprojekteissa. Sisäinen projekti jää lähes aina asiakasprojektien jalkoihin, ja se vaikeuttaa asioiden toteutumista. Asiakasprojekteissa on mahdollisesti enemmän nähtävissä käytettävän ketteriä projektimenetelmiä kuin sisäisissä projekteissa. Tämäkin projekti olisi onnistunut, jos projektin aikataulusta olisi pidetty kiinni. Todennäköisesti se olisi myös rasittanut kaikkia vähemmän. Se olisi hetkellisesti rajoittanut asiakastöiden tekemistä, mutta se olisi saatu nopealla aikataululla valmiiksi, minkä jälkeen kehittäjä ja muutkin projektissa mukana olevat olisivat voineet keskittyä pelkästään asiakastyön tekoon.

6.2 Projektin tulevaisuus

6.2.1 Testaus

Tapauksenhallintajärjestelmän kehittämisprojektin toteutusvaihe on insinööriyön palautushetkellä vielä kesken. Edes ensimmäistä versiota järjestelmästä uusin ominaisuuksin ei ole vielä saatavilla. Kun ominaisuuksien kehitys saadaan käyntiin, on edessä testaus, vanhan datan vieminen uuteen järjestelmään ja käyttöönoton suunnittelu ja toteutus.

Testauksella tarkoitetaan vaihetta, jossa uutta järjestelmää testataan ja jonka tarkoituksena on löytää mahdolliset virheet. Testaus voidaan suorittaa monella eri tavalla. V-malli on yleinen tapa suorittaa testaus. Siinä testaus jaetaan kolmeen eri tasoon: moduuli-, integrointi- ja järjestelmätestaukseen. Kuvassa 17 on kuvattu testauksen v-malli.



Kuva 17: Testauksen v-malli (14).

Moduulitestauksessa testattavana ovat järjestelmän yksittäiset moduulit. Jo määrittelyvaiheessa halutut ominaisuudet jaetaan pieniin moduuleihin, joita tarkastellaan yksittäin. Näin ollen moduulit voidaan testata myös yksittäin. Testauksen suorittaa joko tekninen asiantuntija itse tai yhdessä toisen asiantuntijan kanssa. Moduulitestaus on tärkein vaihe koko testauksessa, joten se on suoritettava huolellisesti. (15.)

Seuraava vaihe testauksessa on integrointitestaus, joka usein saattaa edetä moduulitestauksen rinnalla. Siinä testataan moduulien tai moduuliryhmien yhteistoimintaa ja etsitään siitä virheitä. Yksittäiset moduulit saattavat toimia hyvin yksin, mutta kun ne yhdistetään moduuliryhmään, ne eivät välttämättä toimi halutulla tavalla. Integrointitestauksessa on tärkeää keskittyä moduulien välisten rajapintojen toimivuuteen. Moduuliryhmät voidaan muodostaa joko yhdistämällä kaikki olemassa olevat moduulit kerralla yhdeksi ryhmäksi tai lisäämällä moduuleja vähän kerrallaan ja kasvattaa ryhmää. On kuitenkin hyvä huomioida ryhmiä tehtäessä, että mahdolliset virheet on helpompi huomata pienimmissä ryhmissä. (15.)

Testauksen viimeinen vaihe on järjestelmätestaus, jossa nimen mukaan testattavana on koko järjestelmä. Järjestelmä kattaa muun muassa laitteiston, ohjelmakoodin, tietokannat ja mahdolliset ulkoiset laitteet. Testaus suoritetaan niin kuin järjestelmää käytetään oikeassa tilanteessa. Järjestelmätestauksen tarkoituksena on havaita mahdolliset virheet ohjelman toiminnassa, kapasiteetissa, poikkeuksien toipumisessa tai tietoturvassa. (15.)

Työssä tutkittava järjestelmä pitäisi testata v-mallin mukaisesti. Järjestelmään ohjelmoidaan monia uusia ominaisuuksia, jotka on testattava ensin yksitellen. Kun ominaisuudet on testattu ja todettu toimiviksi yksitellen, niistä pitää muodostaa isompia kokonaisuuksia ja testata niiden yhteensopivuus. Loppujen lopuksi järjestelmää pitää vielä testata kokonaisuutena. Testauksessa on tärkeää, että siihen osallistuu erilaisia henkilöitä. Teknisen asiantuntijan on hyvä testata järjestelmää teknisestä näkökulmasta, asiakaspalvelijan omasta ja palvelupäällikön omasta näkökulmastaan. Asiakkaiden käyttöliittymä täytyy testata ensin yrityksen sisällä, ennen kuin asiakkaat testaavat sitä. Näin voidaan välttää mahdollisten isojen virheiden saattaminen asiakkaiden tietoon.

Tärkeä vaihe testauksessa on myös käytettävyyden testaus. Sen tarkoituksena on seurata käyttäjien reaktioita ja toimintaa heidän käyttäessään järjestelmää ja sitä kautta tehdä havaintoja ja parantaa järjestelmän laatua. Testauksen merkitystä ei pidä kehityksen aikana vähätellä. Kun järjestelmää kehitetään eteenpäin, on syytä muistaa varmistaa järjestelmän käytettävyys. Testausta on syytä tehdä koko kehitysprojektin elinkaaren aikana, koska pienempiä osia on helpompi testata ja niistä on helpompi havaita virheitä kuin yhdestä isosta järjestelmästä. Jos testaus tehdään vasta, kun koko järjes-

telmä on valmis, on suurempi riski virheiden huomaamattomuuteen. Isossa kokonaisuudessa pienet virheet voivat jäädä isojen kokonaisuuksien varjoon, ja tämän takia olisi tärkeää suorittaa testejä koko elinkaaren aikana. (5, s. 297–301.)

Käytettävyystestaus voidaan jakaa kolmeen osaan: ennakkotyöt, testin tekeminen ja testitulosten analysointi ja raportin tekeminen. Testi on tärkeää suunnitella kunnolla ennen sen aloittamista. Testin ennakkotöihin kuuluvat muun muassa seuraavat vaiheet:

- testin tavoitteiden selvittäminen
- käyttäjäryhmien selvittäminen, testikäyttäjämäärän selvittäminen ja rekrytoinnin aloitus
- testattavien toimintojen valinta
- testitarinan ja -tehtävien valinta
- testipaikan valinta
- testausmenetelmien valinta
- mahdollinen interaktiivinen tilanne
- pilottitestin järjestäminen (5, s. 302–305).

Kun ennakkotyöt ja testin suunnittelu ovat valmiit, on edessä testin järjestäminen. Käytettävyystesti etenee yleensä tietyn rakenteen mukaan. Ennen testin aloittamista on hyvä selvittää käyttäjille testitilanne. Monelle se voi olla aivan uusi tilanne, eivätkä kaikki ymmärrä, mitä testitilanteella tarkoitetaan. On hyvä ensin tehdä selväksi, että tarkoitus on testata järjestelmää, ei käyttäjien taitoja, ja että testi on vapaaehtoinen ja käyttäjä voi sen keskeyttää milloin tahansa. Selvityksen jälkeen on käyttäjien alkukysely tai -haastattelu, jonka tarkoituksena on kartoittaa käyttäjien taustaa ja osaamista. Haastattelun jälkeen käyttäjät pääsevät tekemään itse testausta. Testauksen alkaessa on hyvä käydä testitarinan alkutilanne läpi kaikkien käyttäjien kanssa, niin vältetään epäselvyyksiltä. Testaustehtävien jälkeen on vielä kaikille käyttäjille tehtävä loppuhaastattelu, jossa käydään läpi käyttäjän tuntemuksia ja käyttökokemuksia. (5, s. 306–307.)

Testin jälkeen testinpitäjät keräävät saadut materiaalit ja analysoivat ne. Johtopäätökset kirjataan loppuraporttiin. Riippuen testin tarkoituksesta voidaan testin tulos luovut-

taa pelkästään virhelistana tai sitten kokonaisena raporttina. Tyypilliseen raporttiin kootaan seuraavat asiat:

- tuotteen käyttöliittymän ja mahdollisesti käyttötavan lyhyt kuvaus
- lyhyt kuvaus testaustavasta ja testikäyttäjistä
- testattavat toiminnot tai testitehtävät
- testin tulos (virheet tai korjausehdotukset)
- testin ohjaajan lausunto järjestelmästä ja yhteenveto testistä
- yhteenveto virheistä järjestettynä niiden vakavuuden mukaan (5, s. 308–309).

6.2.2 Käyttöönotto

Kun testaus on saatu valmiiksi, seuraava vaihe on käyttöönotto. Ennen kuin järjestelmä voidaan ottaa kunnolla käyttöön, täytyy vanhat tiedot siirtää uuteen järjestelmään. Ennen uutta järjestelmää käytössä on ollut kaksi samankaltaista järjestelmää, ja niistä tuodaan kaikki vanhat tapaukset uuteen järjestelmään. Vanhoissa tapauksissa ei oteta huomioon esimerkiksi vasteaikoja, joten ne tulevat vain historiatiedoksi. Myös vanhan järjestelmän olemassaolo on osa käyttöönottoa. Vanha järjestelmä on jonkin aikaa rinnakkain käytössä uuden järjestelmän kanssa, kunnes kaikki tapaukset on saatu suljetua vanhasta järjestelmästä. Käyttöönottoa suunniteltaessa on tärkeää muistaa koulutuksen merkitys. Kyseessä on aivan uusi järjestelmä kaikille muille paitsi projektissa mukana olleille, joten koulutukset on hyvä sopia hyvissä ajoin heti käyttöönoton alkuun. (2, s. 37.)

Monella saattaa olla harhakuva, että järjestelmän elinkaari loppuisi käyttöönottoon. Melkein voisi sanoa, että käyttöönotosta järjestelmä vasta herää eloon, koska käyttäjät pääsevät käyttämään sitä. Käyttöönoton jälkeen seura ylläpitovaihe, joka on yleensä järjestelmän pisin elinkaaren vaihe. Ylläpitovaiheessa tärkeintä on huolehtia järjestelmän toimintakunnosta, jatkokehityksestä ja muista muutoksista. Ylläpito voidaan jakaa neljään eri perustapaukseen:

- korjaava ylläpito, jossa keskitytään korjaamaan käyttöönoton jälkeen havaitut virheet
- sopeuttava ylläpito, jolla tarkoitetaan järjestelmän muuttamista uusiin ympäristöihin
- täydentävä ylläpito, jossa toteutetaan uusia ominaisuuksia järjestelmään

- ennakoiva ylläpito, jossa keskitytään järjestelmän ja sen dokumentaation parantamiseen. (2, s. 37.)

7 Sisäisten projektien kehitysehdotus ja johtopäätökset

Insinööriyön alkuperäinen ajatus oli tutkia koko tapauksenhallintajärjestelmän kehitysprojektin elinkaarta määrittelystä ylläpitoon asti, mutta työn palautushetkellä projektissa ei ollut päästy määrittelyvaihetta pidemmälle. Päätös järjestelmän kehityksestä tuli elokuun lopussa 2010, ja määrittelyiden teko aloitettiin saman tien. Käyttöönotto piti olla marraskuun alussa 2010. Aikaa kehitystyölle oli alussa varattu kaksi kuukautta.

Määrittelyvaihe saatiin kohtuullisella aikataululla tehtyä, minkä jälkeen kehitystyön piti alkaa. Tässä vaiheessa alkoivat ongelmat. Ensisijainen tekninen asiantuntija joutui täysipäiväisesti työskentelemään asiakasprojektissa, joten täytyi äkkiä löytää toinen järjestelmän teknologiaa osaava asiantuntija, joka löytyi yrityksen toisesta toimipisteestä. Järjestelmä ei ollut hänelle ennestään tuttu, joten aikaa kului järjestelmään tutustumiseen, ja ennen kuin huomasimmekaan asiakastyöt olivat taas kasautuneet ja olimme taas lähtöruudussa. Tässä samassa tilassa projekti on myös työn palautushetkellä.

Mikä siis meni vikaan? Tärkein huomio projektin huonossa etenemisessä on yrityksen sisäinen priorisoinnin puute. Järjestelmän kehitys nähdään tarpeellisena, mutta ei kuitenkaan niin tarpeellisena, että siihen oikeasti panostettaisiin. Järjestelmästä haaveillaan ja puhutaan joka päivä, mutta siltikään sen kehittämistä ei priorisoida korkeammaksi. Yrityksen pitää tietenkin ajatella tuloja, ja niitä saa asiakasprojekteista, mutta jos asiakasprojektit menevät loputtomiin kehitysprojektin edelle, ei järjestelmää saada koskaan käyttöön. Jos työntekijöiden työkalut eivät ole kunnossa, se saattaa pidentää aikavälillä vaikuttaa myös asiakasprojekteihin ja sitä kautta yrityksen tuloihin.

Esimerkkinä järjestelmän tarpeesta on raportointi ja siihen kuluva aika. Tällä hetkellä yrityksessä on käytössä kaksi eri järjestelmää, joista otetaan erilaisia raportteja laskutusta varten. Jo pelkkään raportointiin menee niin kauan, että siihen voitaisiin palkata yksi palvelupäällikkö tekemään sitä kokopäiväisesti. Jos järjestelmän kehitys vietäisiin loppuun, helpottuisi raporttien otto integraation kautta. Näin ollen työntekijät voisivat

olla aktiivisempia asiakkaiden kanssa eikä aina vain tekemässä raportteja, ja kun asiakkaat ovat tyytyväisiä, he myös tuovat yritykselle tuloja.

Yrityksen pitäisi panostaa enemmän sisäisten projektien priorisointiin. Kaikki sisäiset projektit eivät tietenkään ole yhtä tärkeässä asemassa kuin tapauksenhallintajärjestelmän kehitys, mutta projektit pitäisi käydä läpi yksitellen ja päättää oikea prioriteetti ja pitää siitä kiinni. Tähän auttaisi sisäisen projektin kulujen laskeminen ja niiden vertaaminen saatuun hyötyyn. Yritysmailmassa raha lähes aina ratkaisee, joten jos sisäisille projekteillekin saataisiin hinta, se helpottaisi priorisointia. Yrityksen pitäisi myös kiinnittää yleisesti huomiota sisäisten projektien toimittamiseen. Verrattaessa tutkittua projektia eri ketteriin menetelmiin, löytyi jokaisesta jotain yhtäläisyyksiä projektiin. Tämä olisi kuitenkin hyvä keskittää niin, että projekti vietäisiin läpi esimerkiksi Scrum-menetelmän mukaisesti. Ketterien menetelmien hyödyntäminen myös sisäisissä projekteissa on tärkeää, koska ne ovat tärkeä osa yrityksen toimintaa ja kehitystä. Menetelmien käyttö ei tietenkään takaa onnistumista, mutta ohjaa ainakin paremmin maaliin.

8 Yhteenveto

Projektin kulku voidaan jakaa karkeasti neljään osaan: suunnitteluun, käynnistämiseen, toteutukseen ja päättämiseen. Jokainen osa on projektissa, mutta kaikkein tärkein vaihe on suunnittelu. Ilman suunnittelua ei voi toteuttaa yhtään mitään. Suunnitteluvaiheen tarkoitus on saada hyvä vaatimusmäärittely ja projektisuunnitelma. Samalla kun dokumentointia edistetään, on hyvä miettiä myös projektin mahdollisia riskejä ja niiden ennaltaehkäisemistä. Ennen projektin varsinaista käynnistämistä on tärkeää myös suunnitella aikataulu ja resurssit.

Vaatimusmäärittelyn tärkeyttä ei voi korostaa liikaa. Kun tutkii erilaisia elinkaarimalleja, vaatimusmäärittelyn tekeminen on aina ensimmäisenä kohtana järjestelmän elinkaarissa. Työssä tutkittiin kolmea erilaista elinkaarimallia: vesiputousmalli, spiraalimalli ja prototyyppilähestymistapa. Vaatimusmäärittelyn tärkein tarkoitus on saada asiakkaan mielikuva paperille ja siitä eteenpäin toteutettaviksi. Kaikki vaatimukset on hyvä kirjata, mutta on tärkeää muistaa, ettei kaikkea voi saada kerralla. Vaatimusten priorisointi auttaa tässä. Vaatimusmäärittely on syytä tehdä alusta asti huolellisesti, koska mahdol-

listen virheiden korjaaminen myöhemmin on kallista ja aikaa vievää. Työssä tehtiin vaatimusmäärittely käyttöön otettavalle tapaustenhallintajärjestelmälle.

Määrittelyvaiheen kanssa samaa tahtia etenee järjestelmän suunnittelu. Suunnittelu- vaiheessa on tärkeä miettiä, miten määrittelyssä olevat vaatimukset voidaan toteuttaa järjestelmään. Suunnittelussa on tärkeä ottaa huomioon muun muassa järjestelmän toiminnallisuus ja rakenne, käyttöliittymän eri elementit ja visuaalinen suunnittelu. Työssä tutkitun järjestelmän suunnitteluvaihe erosi perinteisestä suunnitteluvaiheesta, koska kyseessä on jo käytössä oleva järjestelmä eikä kokonaan uusi. Järjestelmän toiminnallisuuteen ja rakenteeseen ei projektin aikana koskettu. Käyttöliittymä ja visuaalinen suunnittelu piti ottaa huomioon puuttuvia ominaisuuksia suunniteltaessa.

Kolmesta tärkeimmästä ominaisuudesta, jotka puuttuvat järjestelmästä, tärkein ja liiketoimintakriittisin ominaisuus on vasteaikojen puuttuminen. Tukipalvelusopimukset velvoittavat yrityksen toimimaan tiettyjen vasteaikojen sisällä, ja jos näin ei tapahdu, on yritys mahdollisesti velvollinen maksamaan sanktioita. Toinen puutteellinen ominaisuus on tapausten jononäkymä. Jononäkymässä kuuluu näkyä kaikki avoinna olevat tapaukset statuksesta ja asiakkaasta riippumatta, ja näin ei tällä hetkellä järjestelmässä ole. Kolmas ominaisuus on raportointi, jota palvelupäälliköt ovat velvollisia tekemään ja toimittamaan asiakkaille kuukausittain.

Kun projektin suunnitteluvaihe on valmis, on aika käynnistää projekti kunnolla ja aloittaa vaatimusten toteutus. Toteutusvaiheen tuotos on valmis tuote. Nykyään kuitenkin projektin onnistuminen ei ole itsestään selvä asia. Koko ajan halutaan korkeampilaatuisia järjestelmiä mahdollisimman pienillä kustannuksilla. Tämän ilmiön myötä projekti-toiminnassa ovat yleistyneet erilaiset ketterät toimintatavat. Työssä esitellään neljä yleisintä toimintatapaa: Agile Modeling, Dynamic Systems Development Method, Extreme Programming ja Scrum. Ketterät toimintatavatkaan eivät tosin takaa projektin onnistumista.

Työssä tutkitun projektin tulevaisuus on työntäyteinen. Suurimmat askeleet projektissa ovat vielä muutosten toteuttaminen, järjestelmän testaus, vanhojen tietojen tuominen vanhasta järjestelmästä uuteen järjestelmään ja varsinainen käyttöönotto. Käyttöönoton jälkeen järjestelmän ylläpidosta täytyy myös sopia.

Työssä kehitettävänä olleen järjestelmän kehitys jäi kesken projektin aikataulun venymisen vuoksi, mutta työn aikana saatiin kuitenkin dokumentoinnin tärkein osa valmiiksi. Projektin aikana huomattiin dokumentoinnin ja varsinkin määrittelyiden tärkeys. Jos järjestelmänvaatimukset on hyvin määritetty, järjestelmää on helpompi lähteä kehittämään. Yksin määrittelyllä ei kuitenkaan järjestelmää kehitetä. Työ osoitti hyvin resurssoinnin ja priorisoinnin tärkeyden myös yrityksen sisäisissä projekteissa.

Lähteet

- 1 Saarti, Jarmo. 2000. Projektitoiminta ja kansainväliset hankkeet. Verkkodokumentti. Kreodi. <<http://www.kreodi.fi/artview.asp?ArticleID=76>>. Luettu 15.3.2011.
- 2 Pohjonen, Risto. 2002. Tietojärjestelmien kehittäminen. Jyväskylä: Docendo Finland.
- 3 Kehittämistyön vaiheet ja elinkaarimallit. Verkkodokumentti. <http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elinkaarimallit_asia.htm>. Luettu 15.3.2011.
- 4 Kankaanpää, Timo. 2004. Määrittelydokumentit. Verkkodokumentti. <http://www.cc.puv.fi/~tku/kurssit/Ohjelmiston_maarittely/maarittelydokumentit.html>. Luettu 15.3.2011.
- 5 Sinkkonen, Irmeli, Nuutila, Esa & Törmä, Seppo. 2009. Helppokäyttöisen verkkopalvelun suunnittelu. Helsinki: Tietosanoma.
- 6 Korpimies, Kai. 2007. Sovelluksen elinkaari. Verkkodokumentti. <<http://www.helsinki.fi/~korpimie/ohjelmistotekniikka/elinkaari.html>>. Luettu 26.1.2010.
- 7 Korpela, Jukka K. & Linjama, Tero. 2005. Web-suunnittelu. Jyväskylä: Docendo Finland.
- 8 Kuutti, Wille. 2003. Käytettävyys, suunnittelu ja arviointi. Helsinki: Talentum Media.
- 9 Wiio, Antti. 2004. Käyttäjäystävällisen sovelluksen suunnittelu. Edita: Edita Publishing.
- 10 SQL Server Reporting Services. 2008. Verkkodokumentti. Microsoft. <<http://technet.microsoft.com/en-us/library/ms159106.aspx>>. Luettu 15.3.2011.
- 11 Clinton, Keith. 2010. Agile game development with Scrum. Crawfordsville, Indiana: RR Donnelley.
- 12 Holcombe, Mike. 2008. Running an Agile software development project. Hoboken, New Jersey: John Wiley & Sons.
- 13 Scrum figures and wallpaper. Verkkodokumentti. Mountain goat software. <<http://www.mountaingoatsoftware.com/scrum/figures>>. Luettu 15.3.2011.

- 14 Väkeväinen, Petri. 2001. Testaussuunnitelma. Verkkodokumentti.
<<http://www.soberit.hut.fi/tik-76.115/00-01/palautukset/groups/Osprey/t4/testaussuunnitelma.html>>. Luettu 15.3.2011.
- 15 Toroi, Tanja. 2005. Tietojärjestelmien testaus. Verkkodokumentti.
<his.uku.fi/avointa/julkaisut/TestausluentoESHI111005.ppt>. Luettu 15.3.2011.

Asiakastyytyväisyyskyselyn tulokset

SD&S asiakastyytyväisyyskysely May 2010

Selainpohjainen järjestelmä on hyödyllinen / Web-based system is useful	
Erittäin tyytyväinen / Very satisfied	10
Tyytyväinen / Satisfied	10
Tyytymätön / Unsatisfied	4
Erittäin tyytymätön / Very unsatisfied	0
Helppokäyttöisyys / Easy to use	
Erittäin tyytyväinen / Very satisfied	1
Tyytyväinen / Satisfied	12
Tyytymätön / Unsatisfied	5
Erittäin tyytymätön / Very unsatisfied	5
Visuaalinen ilme / Visual look	
Erittäin tyytyväinen / Very satisfied	2
Tyytyväinen / Satisfied	14
Tyytymätön / Unsatisfied	5
Erittäin tyytymätön / Very unsatisfied	3
Tekninen vakaus / Technical stability	
Erittäin tyytyväinen / Very satisfied	6
Tyytyväinen / Satisfied	11
Tyytymätön / Unsatisfied	5
Erittäin tyytymätön / Very unsatisfied	2
Käyttöohjeistus / Instructions for use	
Erittäin tyytyväinen / Very satisfied	2
Tyytyväinen / Satisfied	13
Tyytymätön / Unsatisfied	6
Erittäin tyytymätön / Very unsatisfied	1

Tiedon löytyminen vanhoista tapauksista / I can find old information on tickets easily	
Erittäin tyytyväinen / Very satisfied	1
Tyytyväinen / Satisfied	8
Tyytymätön / Unsatisfied	9
Erittäin tyytymätön / Very unsatisfied	4
Kommunikoinnin helppous / Ease of communication	
Erittäin tyytyväinen / Very satisfied	2
Tyytyväinen / Satisfied	14
Tyytymätön / Unsatisfied	4
Erittäin tyytymätön / Very unsatisfied	2
Kokonaisarvio / Overall satisfaction	
Erittäin tyytyväinen / Very satisfied	2
Tyytyväinen / Satisfied	12
Tyytymätön / Unsatisfied	7
Erittäin tyytymätön / Very unsatisfied	1

Vasteaikaesimerkin laskukaava

Tapaus kirjataan järjestelmään maanantaina 22.11.2010 kello 9.15. Asiakkaan palvelu-aika maanantaista perjantaihin on 9-17 (Suomen aikaa), pois lukien arkipyhät. Sivulla 24 olevan vasteaikataulukon mukaan asiakaspalvelulla on 24 tuntia aikaa merkitä tapaus vastaanotetuksi ja teknisellä asiantuntijalla on 80 tuntia aikaa korjata ongelma.

päivämäärä	kellonaika	käytettyjen tuntien määrä	
ma 22.11.2010	kello 9.15	8 h	
ti 23.11.2010	kello 9.15	16 h	
ke 24.11.2010	kello 9.15	24 h	asiakaspalvelun pitää merkitä tapaus vastaanotetuksi 24.11.2010 kello 9.15
to 25.11.2010	kello 9.15	32 h	
pe 26.11.2010	kello 9.15	40 h	
la 27.11.2010	-	-	
su 28.11.2010	-	-	
ma 29.11.2010	kello 9.15	48 h	
ti 30.11.2010	kello 9.15	56 h	
ke 1.12.2010	kello 9.15	64 h	
to 2.12.2010	kello 9.15	72 h	
pe 3.12.2010	kello 9.15	80 h	teknisen asiantuntijan pitäisi saada ongelma korjattua 3.12.2010 kello 9.15 mennessä, jotta vasteaika ei ylittyisi vasteaikataulukon mukaan
la 4.12.2010	-	-	
su 5.12.2010	-	-	
ma 6.12.2010	-	-	itsenäisyyspäivä
ti 7.12.2010	kello 9.15	88 h	lisätään aika, jonka tapaus oli odottaa tilassa
ke 8.12.2010	kello 12.25	91 h 10min	lisätyn ajan jälkeen vasteaikaylittyisi ke 8.12.2010 kello 12.25
Huomioitavaa:			
ke 24.11.2010	kello 12.45		asiakaspalvelu kysyy asiakkaalta lisätietoja ja vaste aika pysähtyy
to 25.11.2010	kello 15.55		asiakas vastaa viestiin, vasteaikaa käynnistyy uudestaan ja odottaa-tilassa ollut aika lisätään vasteaikaan (11 h 10 min). Lisäyksen jälkeen kokonaisaika on 91 h 10 min.